



GÖTEBORGS UNIVERSITET

Användarinvolvering för kontinuerlig utveckling i Software as a Service

Utformningen av ett feedback-verktyg

User involvement for continuous development in Software as a Service

The design of a feedback tool

MATTIAS JOHANSSON
CHRISTOFFER LINDBERG

Kandidatuppsats i Informatik

Rapport nr. 2015:033

Abstrakt

Något som blivit mer populärt på senare tid är att hyra eller prenumerera på IT-tjänster, så kallad Software as a Service (SaaS). Att kunderna inte behöver göra stora investeringar vid införskaffandet av IT-system innebär att de, utan några större förluster, snabbt kan byta leverantör av system. Därför är det viktigt att upprätthålla en hög användaracceptans för att inte förlora kunder. Med hjälp av användarinvolvering kan företag som tillhandahåller SaaS-tjänster öka sina chanser att ge en god service och behålla sina kunder. Ett sätt att involvera användarna i system som är i drift, är genom inhämtning av feedback. Vår forskningsfråga *“Vilka designmönster kan användas för att skapa en miljö i ett SaaS-system som gör att användarna enkelt kan ge kvalitativ feedback?”* har vi besvarat med hjälp av metoden Action Design Research. Våra resultat indikerar att Clear Entry Point, Modal Panel och Escape Hatch alla är designmönster som med fördel kan användas. Utöver detta har vi presenterat en “drag and drop”-lösning på problemet med att användare ska beskriva var i systemet de avser att beskriva något.

Nyckelord: Software-as-a-service, SaaS, användarinvolvering, användarmedverkan, feedback, designmönster

Abstract

To rent or to subscribe to IT services, so called Software as a Service (SaaS), is something that has become more popular recently. The fact that customers do not need to make large investments for procurement of IT systems means that they, without any major losses, can quickly change supplier of system. It is therefore important to maintain a high user acceptance in order not to lose customers. With the help of user involvement companies providing SaaS services can increase their chances to provide a good service and thereby keep their customers. One way to involve the users in operational systems is collecting feedback. Our research question "Which design patterns can be used to create an environment in a SaaS system that allows users to easily provide qualitative feedback?" is answered with the method Action Design Research. Our results indicate that Clear Entry Point, Modal panel and Escape Hatch are all design patterns that can be advantageously used. In addition to this, we have presented a drag and drop solution to the problem that users have to describe where in the system they intend to describe something.

The report is written in Swedish.

Keywords: Software as a service, SaaS, user involvement, user participation, feedback, design patterns

Tack

Vi vill rikta ett stort tack till Vendium AB för ett väldigt gott samarbete.

Vi vill också tacka vår effektive och kunnige handledare Johan Magnusson.

Innehåll

1	INTRODUKTION	1
1.1	BAKGRUND	1
1.2	PROBLEM	2
1.3	SYFTE OCH FRÅGESTÄLLNING.....	2
1.4	DEFINITION OCH AVGRÄNSNING.....	2
2	BEGREPPSRAM OCH TIDIGARE FORSKNING	3
2.1	AFFÄRSMODELLEN SOFTWARE-AS-A-SERVICE.....	3
2.2	ANVÄNDARINVOLVERING	3
2.2.1	<i>Inhämtning av användar-feedback i system i drift.....</i>	<i>5</i>
2.2.2	<i>Reducera barriären för användare att delta.....</i>	<i>6</i>
2.3	DESIGNMÖNSTER.....	7
2.3.1	<i>Navigation.....</i>	<i>7</i>
2.3.2	<i>Organisering av layout.....</i>	<i>8</i>
3	METOD	9
3.1	EMPIRISKT URVAL	9
3.1.1	<i>Fallstudieobjekt.....</i>	<i>9</i>
3.1.2	<i>Urval.....</i>	<i>9</i>
3.2	ACTION DESIGN RESEARCH.....	9
3.3	VALET AV ADR.....	12
3.4	VÅR TILLÄMPNING AV ADR	12
3.4.1	<i>Fas 1, problemformulering</i>	<i>13</i>
3.4.2	<i>Fas 2, UIE.....</i>	<i>13</i>
3.4.3	<i>Fas 3, reflektion och lärande.....</i>	<i>14</i>
3.4.4	<i>Fas 4, formalisering av lärande.....</i>	<i>14</i>
3.5	DATAINSAMLING	14
3.5.1	<i>Intervjuer.....</i>	<i>14</i>
3.5.2	<i>Intervju med informant 1</i>	<i>15</i>
3.5.3	<i>Intervjuer med slutanvändare</i>	<i>15</i>
3.5.4	<i>Evaluering.....</i>	<i>15</i>
4	RESULTAT	17
4.1	PROBLEMFORMULERING	17
4.1.1	<i>Feedback-inhämtningen innan vår inblandning</i>	<i>17</i>
4.1.2	<i>Krav.....</i>	<i>18</i>
4.1.3	<i>Lösningsförslag</i>	<i>18</i>
4.1.4	<i>Slutsats.....</i>	<i>19</i>
4.2	UTVECKLA, INGRIPA OCH EVALUERA	19
4.2.1	<i>Iteration 1.....</i>	<i>19</i>
4.2.2	<i>Iteration 2.....</i>	<i>21</i>
4.2.3	<i>Iteration 3.....</i>	<i>26</i>
4.2.4	<i>Slutsats.....</i>	<i>30</i>
4.3	REFLEKTION OCH LÄRANDE	31
5	DISKUSSION	32
5.1	BEGRÄNSNINGAR	35
6	SLUTSATS.....	36
6.1	STUDIENS RELEVANS OCH ÖVERFÖRBARHET	36
6.2	FÖRSLAG TILL VIDARE FORSKNING	36
7	REFERENSER.....	37

Bilaga 1 – Intervjuguide intervju 1, informant 1

Bilaga 2 – Intervjuguide telefonintervju 1, informant 1 och 2

Bilaga 3 – Evalueringinstruktion för informant 2 och 3

Bilaga 4 – Evalueringsteman för iteration 2 och 3 med informant 1

Bilaga 5 – Intervjufrågor evaluering med informant 2 och 3

Bilaga 6 - Intervjuförfrågan

1 Introduktion

I detta avsnitt behandlar vi bakgrunden till problemområdet och problematiken som vi med hjälp av vårt syfte samt frågeställning ska ge svar på. Sedan avslutas avsnittet med att klargöra vissa definitioner för att förenkla vidare läsning av denna uppsats.

1.1 Bakgrund

I stället för att på traditionellt vis köpa ett nytt system som hanteras på egen server så kan man nu hyra system eller tjänster från en tredje part, så kallad Software as a Service (SaaS). Det är något som har blivit allt mer populärt på senare tid (Magnusson & Nilsson 2014). Företagen besparas från att lägga tid på förvaltning av servrar och underhåll av system och kan fokusera på sin kärnverksamhet (Benlian et al. 2009). Eftersom man oftast (beroende på prenumerationsmodell) snabbt kan avsluta sitt användande av systemet och övergå till något annat, ställer det högre krav på användaracceptans (Vella 2011).

Genom att involvera användarna i utformning och utveckling av system så kan man öka acceptansen hos användare (Damodaran 1996; McKeen et al 1994). En förklaring till den ökade acceptansen är att användaren får en känsla av kontroll när de själva kan vara delaktiga i utformningen av systemet (Kujala 2003). Begreppet användarinvolvering innefattar alla olika arbetssätt som innebär kontakt med användarna. Detta kan innebära att användare tar emot eller förser utvecklarna med information, ger synpunkter på fördefinierade delar eller är medverkande vid design (Damodaran 1996).

Att involvera användarna när ett system är i drift är viktigt för organisationer som vill hålla sig konkurrenskraftiga och sikta på en god service gentemot sina användare, speciellt för webbaserade tjänster där konkurrensen är mer direkt (Bragge & Merisalo-Rantanen 2007). Ett sätt att involvera användarna när ett system är i drift är att inhämta feedback när de använder systemet. För att öka användaracceptansen vid utveckling, bör designen vara utformad så att användaren känner igen sig och att man använder sig av prövade och testade designelement, detta för att användaren lättare ska komma igång med användningen (Sharp et al 2010). Dessa prövade och testade designelement finns samlade som designmönster (Tidwell 2011).

1.2 Problem

Det finns mycket forskning kring användarinvolvering, främst fokuserar den på användarinvolvering i ett tidigt stadium vid utveckling av system (se t.ex. Kujala 2003; Alam 2002; Damodaran 1996).

Forskning kring användarinvolvering av system i bruk finns det mindre av (se t.ex. Bragge & Merisalo-Rantanen 2007). Litteratur som beskriver utformning av en integrerad feedback-funktion består främst av konferensartiklar (se t.ex. Lohmann & Rashid 2008; Heller et al 2011; Schneider 2011) och är därför inte så utförlig, dessutom så riktar sig ingen av dessa till SaaS-system.

1.3 Syfte och frågeställning

Syftet med vårt arbete är att utforska hur man kan designa en komponent för att inhämta kvalitativ feedback från användare av SaaS-system, för att på så vis förbättra användarinvolveringen.

Undersökningen har gjorts i samverkan med ett företag samt deras användare med den vetenskapliga metodiken Action Design Research (ADR). Frågeställningen vi arbetat mot genom hela projektet och således baserat vår undersökning på är:

”Vilka designmönster kan användas för att skapa en miljö i ett SaaS-system som gör att användarna enkelt kan ge kvalitativ feedback?”

1.4 Definition och avgränsning

När vi skriver ”feedback” så menar vi åsikter från användaren kring något som berör systemet, inte Normans (2002) definition av feedback eller andra designmönster. Vi har laborerat med andra uttryck likt respons och återkoppling men finner feedback som det som beskriver det vi menar mest intuitivt.

Eftersom designmönster är ett brett begrepp vill vi klargöra att när vi skrivit designmönster i uppsatsen så menar vi Tidwells (2011) designmönster för utformning av grafiska element.

I arbetet används begreppet användarinvolvering som en generell term som innefattar alla olika arbetssätt som innebär kontakt med användarna (Damodaran 1996). Vi menar inte Barki och Hartwicks (1994) definition av begreppet, som innebär hur känslomässigt involverad användaren är över ett system.

Systemet som vi gjort vår artefakt i är webbaserad och anpassad för åtkomst via dator och mus, vi har inte gjort någon version för navigering med touch. Vi har inte heller gjort något ingripande i deras mobila applikationer.

2 Begreppsram och tidigare forskning

I detta avsnitt avhandlas den relaterade teorin som vi grundar vårt arbete på. Dessa är affärsmodellen Software-as-a-Service, användarinvolvering, inhämtning av feedback samt designmönster.

2.1 Affärsmodellen Software-as-a-Service

SaaS är en affärsmodell som bygger på att mjukvaruleverantörer tillgängliggör för kunder en mjukvaruapplikation i webbläsaren via internet (Wu 2011; Verma 2011). Dessa tjänster har stått för ett paradigmskifte i mjukvaruanvändandet bland företag det senaste årtiondet, där användare nyttjar tredjeparts, internetbaserade mjukvaruapplikationer för sina datalagrings- och dataprocesserande behov istället för mer traditionella Commercial-off-the-shelf-applikationer (COTS) installerade på deras datorer (Bhattacharjee & Park 2014). SaaS kan ses ur två olika perspektiv, dels ur perspektivet för de som tillhandahåller tjänsten (leverantörer) och dels för de som nyttjar tjänsten (kunder eller användare) (Papazoglou & Ribbers 2006; Armbrust et al. 2010).

Inkomsterna för SaaS-leverantörerna kommer vanligtvis från antingen tidsavgränsade prenumerationer eller beroende på hur mycket de använder tjänsten (Verma 2011).

Benlian (2011) menar att mjukvara utvecklad för SaaS har drastiskt lägre kostnader gentemot COTS-mjukvarumodeller på grund av att leverantörer och utvecklare ges möjlighet att ha full kontroll över hela mjukvarustacken. Dessa kostnadsfördelar ärvs också av kunder som drar nytta av lägre totala ägandekostnader. Den största fördelen med on-demand-lösningar är de relativt snabbt och enkelt kan "installeras", anpassas och användas av kunderna (ibid). En annan skillnad mellan SaaS-mjukvara och traditionella är att uppdateringar kommer i mindre men mer frekventa stötar inom SaaS (Dubey & Wagle 2007).

Leverantörer av SaaS är likt de flesta företag beroende av sina kunder och att de är nöjda. Skillnaden i att hålla sina kunder nöjda för SaaS-leverantörer gentemot COTS är att kunderna på grund av lägre uppstarts- och ägandekostnader har kunderna mycket lättare att byta leverantör om den upplevs som undermålig (Dubey & Wagle 2007). Detta ställer höga krav på användaracceptans (Vella 2011).

2.2 Användarinvolvering

Användarinvolverad utveckling av IT-system är vida accepterat vid skapandet av användbara system. Forskning visar att användarinvolverad utveckling ger system en ökad kvalitet, den visar också att användarinvolvering ger en ökad acceptans för samt förståelse av systemet och att det slutligen ger en effektivare användning av systemet (Damodaran 1996). En förklaring till den ökade acceptansen

är att användaren får en känsla av kontroll när de själva kan vara delaktiga i utformningen av systemet (Kujala 2003). Men användarinvolvering är ett brett begrepp som man i forskning definierat på olika sätt. Ives och Olson (1984) definierar användarinvolvering som att användarrepresentanter från den tänkta målgruppen medverkar i systemutvecklingsprocessen. Men Barki och Hartwick (1994) vill skilja på begreppet användarinvolvering (user involvement) och användarmedverkan (user participation). De definierar användarinvolvering som den personliga betydelse ett system har för användaren, medan de definierar användarmedverkan som en samling aktiviteter utförda av användarna i systemutvecklingsprocessen. I arbetet så används begreppet användarinvolvering som en generell term som innefattar alla olika arbetssätt som innebär kontakt med användarna, detta enligt Damodaran (1996), som gått igenom tidigare forskning inom området och menar att former av användarinvolvering brett kan karakteriseras att hamna någonstans i spannet som representeras i tabell 1.

Informativ	Konsultativ	Medverkande
Användare förser och/eller tar emot information	Användare kommenterar på fördefinierade tjänster	Användare influerar beslut relaterat till hela systemet

Tabell 1. Damodaran (1996) olika former av involvering.

I tidigare forskning har Ives och Olson (1984) en annan klassificering av användarinvolvering, det är graden av inflytande användaren har över slutprodukten. Den innehåller sex stycken grader.

Ingen involvering: Användaren vill inte eller är inte inbjuden att delta.

Symbolisk involvering: Användarens åsikt är ombedd men ignorerad.

Rådgivande involvering: Råd och åsikter är ombedda genom intervjuer eller enkäter.

Involvering med svag kontroll: Användarna har ansvaret att godkänna varje steg i en utvecklingsprocess.

Praktisk involvering: Användaren är medlem av designteamet, eller är officiell samarbetspartner med utvecklingsgruppen.

Inblandning med stark kontroll: Användarna betalar för utveckling ur egen ficka.

Damodaran (1996) tre former av användarinvolvering och Ives och Olsons (1984) sex grader kan ses som två dimensioner av användarinvolvering som delvis överlappar varandra, dels på vilket sätt det utförs och dels vilket inflytande det har på systemet. I Damodaran (1996) olika former är medverkande den enda formen som visar att användaren har något inflytande kring ett system. I formerna informativ och konsultativ så tar man emot information/åsikter från användarna men det behöver inte betyda att det i slutändan har inflytande på systemet. I en undersökning av Lynch och

Gregor (2004) där man tittade på 38 systemutvecklingsprojekt av beslutstödssystem som använde sig av användarinvolvering, så fann man att det är graden av inflytande användarna har som har störst påverkan om ett system lyckats, inte om de använder sig av användarinvolvering.

En stor andel av forskningen inom användarinvolvering lägger vikt vid att den ska ingå i ett tidigt stadium i systemutvecklingsprocessen för ett lyckat systemutvecklingsprojekt (Kujala 2003; Alam 2002; Damodaran 1996). Men Bragge och Merisalo-Rantanen (2007) anser även att användarinvolvering ska ske när systemet är i drift. Detta menar de är ett måste för organisationer som vill hålla sig konkurrenskraftig och sikta på att ha en god service gentemot sina användare, speciellt inom webbaserade tjänster och informationssystem där det existerar direkt konkurrens. Att samla in feedback från användarna i ett system i drift anser Schneider (2011) är väsentligt för att stödja evolutionen av systemet.

2.2.1 Inhämtning av användar-feedback i system i drift

Ett sätt att involvera användarna när ett system är i drift är att hämta in feedback från dem när de använder sig av systemet. Nyttan med att ta in feedback från användarna när systemet är i bruk är bland annat att avslöja missförstånd mellan leverantör och slutanvändare (Schneider 2011).

Användar-feedback innehåller viktig information för utvecklare. Bland annat så hittas många buggar och användbarhetsproblem först när systemet är i bruk (Heller et al. 2011). Med det så ökar kvaliteten på systemet, dessutom så kan man identifiera avsaknaden av funktioner (Pagano & Bruegge 2013). För att motivera användarna till att lämna feedback så föreslår Nichols och Twidale (2003) att man låter användaren få veta att och hur feedbacken har påverkat utvecklingen av systemet. Vidare visar Pagano och Bruegge (2013) i sin studie att implementering av en funktion som önskats av användare kan leda till att användarna går ut offentligt och hyllar företaget, vilket kan få till följd att potentiella kunder får upp ögonen för produkten.

För att kunna hämta in feedback av sina användare i ett system som är ute på marknaden finns olika sätt att tillgå. Lohman och Rashid (2008) nämner två olika dimensioner för fjärrdeltagande, självstyrande eller händelsedrivna. Självstyrande deltagande är när användaren rapporterar feedback spontant när den känner för det inne i systemet. Händelsedrivna feedback är när systemansvariga uttryckligen inbjuder användarna till att ge feedback vid en speciell tid. Bragge och Merisalo-Rantanen (2009) menar att ett självstyrande deltagande gör att bara de mest nöjda eller mest missnöjda användarna är motiverade nog för att lämna feedback. De tillägger också att systemet måste vara av kritisk vikt för användaren för att de ska känna sig motiverade att ge feedback.

Insamlandet av feedback kan ske på olika sätt. Man kan samla in information via olika kanaler som telefon, mail, sociala medier eller integrerade funktioner i systemet. Pagano och Bruegge (2013) menar att nackdelen med att ha spridda kanaler för att inhämta feedback är att informationen ofta kopieras från ett medium till ett annat, vilket kan göra så att möjligheten till att återkoppla till användaren går förlorad. Vidare menar författarna att rapportering av fel inte är så hjälpsamt när det rapporterats in via mail eftersom viktig kontextuell information saknades.

Att ha en feedback-funktion integrerad i systemet är att föredra för att göra det enkelt för användaren att rapportera in något när tanken uppkommer (Nichols et al 2003; Lohmann & Rashid 2008; Pagano & Bruegge 2013). Något som anses hjälpsamt för att förstå kontexten av användar-feedback är att veta vilken funktion som feedbacken åsyftar (Pagano & Bruegge 2013). Lohmann och Rashid (2008) har funnit i sin studie möjligheten att direkt referera till grafiska element under tiden de skriver ner feedback är högt värderat av användare.

När ett system är i bruk, ställs användarna inför situationer som kan leda till spontana klagomål eller förslag på nya funktioner. Dessa fångas in och kan leda till nya krav för utvecklingarna för vidareutveckling av system. En viktig aspekt för att fånga in denna feedback är att sänka tröskeln för användaren att skriva och lämna spontan feedback (Schneider 2011).

2.2.2 Reducera barriären för användare att delta

Lohmann och Rashid (2008) har tagit fram fyra nyckelaspekter som de anser är värdefulla för att lyckas med att minska barriären för att vilja delta samt att bättre länka användar-feedback till funktioner i systemet. Dessa är:

Integration i användarens miljö: Gränssnittet ska i bästa fall vara direkt inbäddat i användarens systemmiljö för att alltid påminna användare att involvering är möjligt, och med det att förbättringar av systemet är möjliga, till exempel någon deltagarknapp som är konstant synlig på skrivbordet eller att den är integrerad i applikationen.

Lättviktig/enkel delaktighet: Det ska vara möjligt för användarna att direkt delta när de får en idé. Detta ska resultera i ett marginellt avbrott i dennes aktuella aktivitet eller arbetsflöde. I bästa fall ska användaren få bestämma vad och hur mycket information den vill delge. Inmatningen ska vara lättviktig och en informell process som senare kan förfinas och utvecklas.

Enkelhet och assistans: All interaktion med användargränssnittet bör vara så simpelt och självförklarande som möjligt för att uppmuntra användarna att vilja involvera sig. Gränssnittet ska inte kräva att man loggar in varje gång en användare vill lämna feedback. Det ska innehålla lämpligt stöd vid interaktion, som automatisk formulärfyllning eller systemförslag. Användaren ska inte ha

krav på sig att förstå systemet med mycket data eller göra klassifikationer som kan vara kognitivt krävande. Men att ta i beaktande är att för mycket assistans kan ha en negativ inverkan på användarens kreativitet.

Transparens: I varje situation måste det vara klart för användaren vilken data som samlas in i och med dennes input. Det ideala är om användaren kan följa upp sin feedback i utvecklingsprocessen. Användarens motivation är starkt baserad på tron att systemet kan förbättras med dennes input.

2.3 Designmönster

Mönster är beteendemässiga och strukturella kännetecken som hjälper till att förbättra vanan och igenkännandet av något. De kan vara en beskrivning av praxis hur man bäst designar en viss domän (Tidwell 2011). Tidwell (2011) har samlat ihop och strukturerat upp designmönster för olika situationer man kan tänkas komma i kontakt med när man designar en artefakt inom IT. Sharp et al (2010) menar att det är viktigt att användaren känner igen sig när de använder en applikation för att lättare komma igång och arbeta i det. Så till vår hjälp har vi gått igenom Tidwells (2011) olika designmönster och plockat ut de som är av särskilt intresse för vårt arbete.

2.3.1 Navigation

Clear Entry Points

Clear Entry Points är ett designmönster som innebär att man visar få ingångar till att börja använda sig av en applikation/funktion. Ingångarna ska vara tydliga och beskrivande.

Escape Hatch

Designmönstret *Escape Hatch* går ut på att det ska vara tydligt för användaren hur den ska kunna navigera ifrån vyer som har begränsande navigationsval. Detta kan göras med att placera en knapp eller en länk som tar användaren tillbaka till där den var innan. Detta designmönster gör att användarna känner sig säkrare när de utforskar en applikation/funktion. Tidwell (2011) rekommenderar detta om man använder sig av designmönstret Modal Panel.

Modal Panel

Modal Panel innebär att det visas en sida med inga andra navigationsval än att användaren avslutar denna uppgift antingen med att stänga ner eller att göra klart uppgiften som visas. Detta mönster används vid visning av meddelanden, ifyllning av formulär och visas oftast som ett lager ovanpå andra sidor.

2.3.2 Organisering av layout

Collapsible Panels

Collapsible Panels är ett designmönster som handlar om att sätta sekundär eller valbar information i paneler som kan öppnas och stängas av användarna, detta för att det inte ska uppta onödig skärmyta.

Movable Panels

Movable Panels är ett designmönster som innebär att man kan flytta runt paneler fritt på sidan, detta används främst vid dashboardapplikationer. Det används när utvecklare inte lägger fokus på var panelen är utan användaren får själv välja utifrån dess egen preferens.

Module Tabs

Mönstret innebär att man sätter innehåll i moduler som sedan klickas fram via små flikar. Bara en modul är synlig åt gången. Detta används när man har mycket information som ska visas och användaren behöver bara se ett innehåll åt gången. Detta mönster är allmängiltigt och användare förstår hur det ska användas (Tidwell 2011).

3 Metod

I detta avsnitt kommer vi presentera vårt empiriska urval, beskriva hur ADR är uppbyggt och hur vi har använt det i vårt arbete. Det ges även en beskrivning av datainsamlingen.

3.1 Empiriskt urval

I följande två avsnitt beskrivs urval av fallstudieobjekt och urval av de informanter vi haft.

3.1.1 Fallstudieobjekt

Eftersom syftet med vårt arbete är att utforska hur man kan designa en komponent för att inhämta kvalitativ feedback från användare av SaaS-system. Valde vi att samarbeta med företaget Vendium AB som var i behov att förbättra feedback-inhämtningen i sitt SaaS-system Struqtur. Struqtur är ett projektverktyg främst inriktat för hantverkare inom byggbranschen, det har varit i drift i 1,5 år och har ungefär 800 användare. Denna tjänst har inte någon bindningstid, vilket gör att det ställs höga krav på systemet, både funktionellt och användarvänlighet. Kunder kan ifall de inte är nöjda snabbt säga upp sin prenumeration.

3.1.2 Urval

Ett kritiskt element enligt Sein et al (2011) är att fastställa en långsiktig relation med den deltagande organisationen. Vi etablerade tidigt en relation med vår kontaktperson på Struqtur (informant 1) som har haft lite olika roller under arbetets gång, beskrivet här nedan. Vi tog också kontakt med två slutanvändare (informant 2 och 3) i iteration 1 för inhämtning av problemformuleringen och för att skapa en kontakt för betatesten som utfördes.

Informant 1: Produktchef och utvecklare på Struqtur.

Som tidigare nämnt så har informant 1 haft olika roller i arbetet. I ett tidigt skede var rollen kontaktperson på Struqtur och i den rollen så diskuterade vi och kontaktpersonen vad vi skulle titta på för slags problem de hade. När problemvalet var gjort har informant 1 använts som dels en rent informativ källa om Struqturs miljö och dels som medutvecklare i form av att vi har bollat idéer.

Informant 2: Administrativ chef och projektledare på ett byggföretag som är kund hos Struqtur.

Informant 3: VD på ett byggföretag som är kund hos Struqtur.

3.2 Action Design Research

I vårt arbete har vi använt oss av Action Design Research (ADR), som enligt Sein et al. (2011) hanterar två till synes olika utmaningar: (1) adressera ett problemområde i en specifik organisatorisk miljö genom att ingripa och evaluera och (2) utveckla och evaluera en IT-artefakt som adresserar den klass

av problem som präglar problemområdet. Svaret på dessa utmaningar resulterar i en metod som fokuserar på utvecklandet, ingripandet och evalueringen av en artefakt, som inte endast reflekterar tidigare teori och forskarnas uppsåt utan också användarnas influenser och användande i sammanhanget (Sein et al. 2011).

Arbetet delas in i sju principer som i sin tur struktureras in i fyra faser: (1) Problemformulering, (2) utveckla, ingripa och evaluera, (3) reflektion och lärande och (4) formalisering av lärande. Dessa faser är iterativa (Sein et al. 2011). Nedan följer en beskrivning av ADR vilken är hämtad från Sein et al. (2011).

Första fasen, problemformuleringen

Den första fasen handlar om att hitta vad problemet faktiskt är. Problemet grundar sig i praktiken eller ett för forskarna förväntat problem. Principerna den innefattar grundar sig dels i praktiken och dels i teorin.

Princip 1: Praktikinspirerad undersökning

Denna princip betonar att praktiska problemsituationer borde ses som kunskapsskapande möjligheter. ADR-teamet bör söka kunskap som kan appliceras på en klass av problem snarare än den specifika situationen. Som ett resultat av detta så är forskningsaktiviteten probleminspirerad.

Princip 2: Teoribaserad artefakt

Denna princip betonar att skapandet av artefakten sker i samverkan med teorier. Man identifierar teorier som passar problemsituationen samt undersöker tidigare liknande teknologiska framsteg. I principen ingår tre överlappande användningar av tidigare teorier: (1) För att strukturera upp problemet, (2) för att identifiera lösningsmöjligheter samt (3) användas som guide vid design.

Andra fasen, UIE (Utveckla, ingripa och evaluera)

I denna fas använder man problemformuleringen och den teoretiska bakgrunden som antagits i den tidigare fasen. Dessa premisser ger en plattform för att kunna generera den initiala designen av IT-artefakten, denna design är sedan formad av organisationens användande och efterföljande iterationer.

Princip 3: **Ömsesidigt skapande** är en princip som tar hänsyn till de delade influenser som IT-artefakten och det organisatoriska sammanhanget har på varandra. Det är en iterativ process där ADR-teamet kan använda sin skapade design av IT-artefakten för att tolka den organisatoriska miljön, för att sedan använda den ökade förståelsen av miljön för att influeras och omdesigna. Beroende på vilken form av UIE man väljer så kan startpunkten för dessa iterationer och forskarnas hållning ändras.

Princip 4: **Ömsesidigt inflytelserika roller** är en princip som pekar på vikten av ömsesidigt lärande mellan de olika deltagarna i projektet. ADR-forskarna har med sig kunskapen från teorin samtidigt som utvecklare har med sig tankar och kunskap från organisationens arbetssätt.

Princip 5: **Autentisk och samverkande evaluering** betonar huvudkaraktäristiken i ADR att evalueringen inte är en separat process som efterföljer utvecklandet utan ska sammanvävas med utvecklingsprocessen. Evalueringsiterationer för alfautförandet är formativa och bidrar till förfiningen av artefakten och synliggör både väntade och oväntade konsekvenser. Senare evaluering av betaversioner är summativa som utvärderar värdet och användbarheten.

Tredje fasen, reflektion och lärande

Fokus förflyttas från byggandet av en lösning för en specifik situation till att applicera lärandet av denna till en bredare klass av problem. I denna fas sker en medveten reflektion av inramningen av problemet, valda teorier och den utvecklade artefakten. Detta är kritiskt för att säkra att bidrag till kunskap är identifierad. Fasen innehåller enligt Sein et al (2011) tre steg. (1) Att reflektera kring designen och omdesignen. (2) Att evaluera att artefakten överensstämmer med principerna. (3) Att analysera resultatet från ingripandet jämfört med sagda mål.

Princip 6. Styrk emergens

Principen betonar att den skapade artefakten inte bara speglar den preliminära designen som skapats av forskarna, utan att den fortlöpande formas av organisationens användande, perspektiv och deltagare samt genom resultatet av autentisk samverkande evaluering.

Fjärde fasen, formalisering av lärande

Syftet med fjärde fasen är att formalisera lärandet som tillkommit i forskningen. Den situationsspecifika kunskapen ska abstraheras till lösningskoncept för en klass av problem i praktiken (i motsättning till teoretiska problem). Forskarna visar på prestationer åstadkomna av artefakten och beskriver de organisationella resultaten för att formalisera lärandet.

Princip 7: Generaliserat resultat

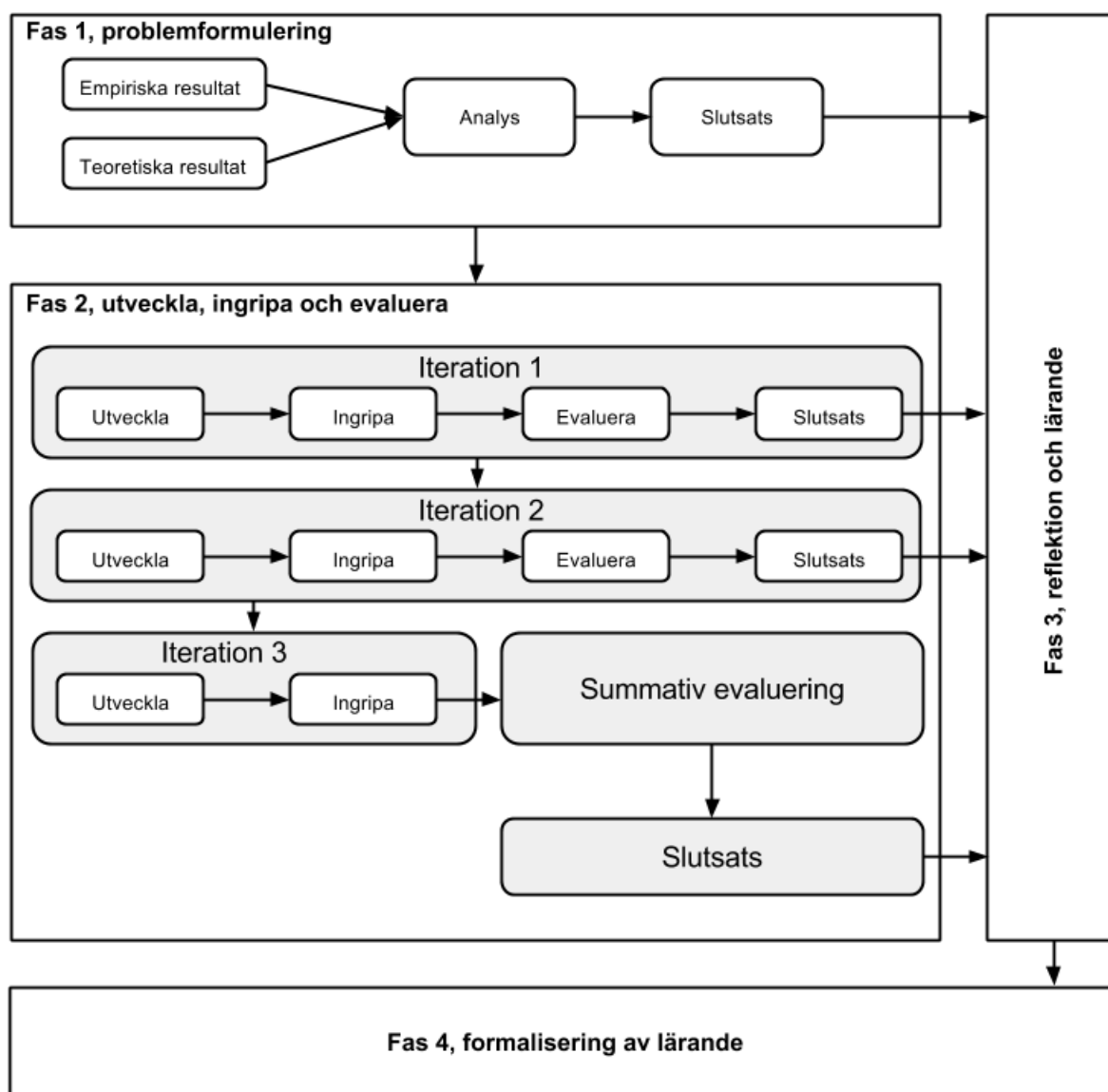
Det kan vara svårt att generalisera de situationsspecifika resultat som ADR medför som inkluderar artefakten och den organisationella miljön. Resultaten är per definition en samling egenskaper inom olika områden. Denna samling representerar lösningen som ska stävja ett problem, båda dessa kan generaliseras. Abstraktionen från specifikt och unikt till generellt är en kritisk del av ADR. Tre nivåer: (1) generalisering av problemet, (2) generalisering av lösningen och (3) härledandet av designprinciper från forskningsresultatet.

3.3 Valet av ADR

ADR är en metod som tillåter oss att dels hjälpa vårt studieobjekt med deras specifika problem och dels bidra till forskningen genom att generalisera det till en klass av problem vi ställts inför och på så vis presentera lösningsförslag på klassen. Informant 1 visade tidigt intresse för att vara med i utvecklingsarbetet. ADR gav oss möjligheten att dra nytta av informant 1:s erfarenhet och kunnande.

3.4 Vår tillämpning av ADR

Nedan följer en beskrivning av hur vi har applicerat ADR i vårt arbete. Beskrivningen presenteras i form av genomgång av de olika faserna och de sju principer som presenterades i avsnitt 3.2.



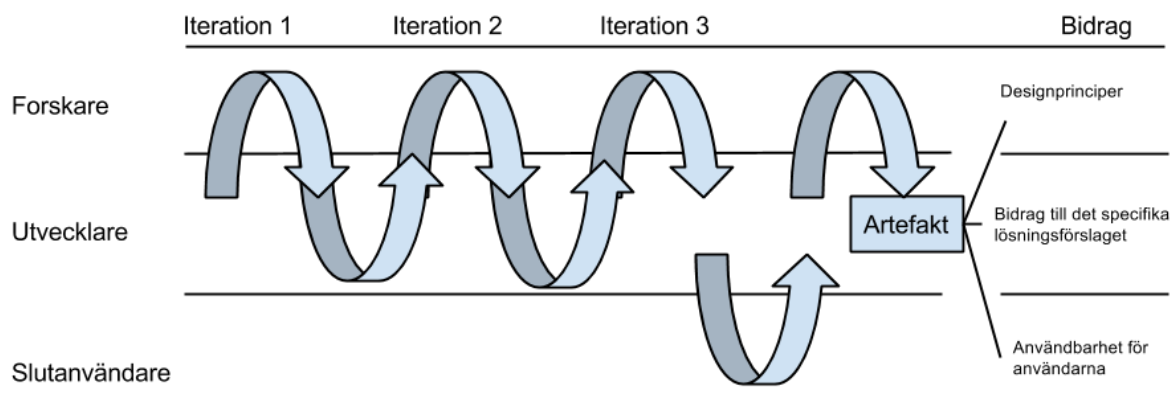
Figur 1. Modell av vår tillämpning av ADR.

3.4.1 Fas 1, problemformulering

Med hjälp av intervjuer och diskussioner med våra informanter samt genomgång av litteratur identifierades generella problematiseringar inom området som lyfter fram viktiga aspekter i hanteringen av feedback. Vi har, enligt princip 1, tittat på problemet ur en generell vinkel där vi har lyft problembilden till något som kan appliceras på liknande fall, en så kallad klass av problem. Som Princip 2 förespråkar så skapade vi vår artefakt med hjälp av de designmönster vi presenterat i avsnitt 2.3.

Roller i arbetet

Nedan följer en beskrivning av rollfördelningen i arbetet. Tre huvudsakliga roller har deltagit och de är indelade i tre huvudsakliga grupper (se figur 2), utvecklare (researcher(s), utvecklare (practitioners) och slutanvändare (end-users).



Figur 2. Beskrivning av förhållandet mellan de olika rollerna i arbetet och iterationsprocessen.

Vi har antagit två roller i arbetet, dels som forskare och dels som utvecklare. I rollen som forskare har vi studerat tidigare arbeten och hämtat inspiration från både litteratur och andra tekniska artefakter. Detta i syfte att berika oss inom problemområdet och utformningen av artefakten. I rollen som utvecklare har vi tillsammans med Informant 1 hittat lösningar på de problem som uppdagats i problemformuleringen, utvecklat vår artefakt, testat artefakten i en testmiljö av systemet för att sedan evaluera den och omvärdera problemformuleringen i de antal iterationer vi tyckte oss behöva. Slutligen har vi låtit slutanvändarna, informant 2 och informant 3, testa artefakten i testmiljön.

3.4.2 Fas 2, UIE

Utifrån informationen vi samlat på oss från fas 1 så började vi tillsammans med Informant 1 att fundera på hur feedback-processen skulle kunna se ut. Utifrån det så började vi skissa på en alfaversion som sedan sattes in i en testmiljö som liknar studieobjektets nuvarande system för att kunna evalueras gemensamt av utvecklarna. Denna fas itererades sedan för att slutligen evalueras

med slutanvändarna. I och med att vi har utvecklat och evaluerat artefakten i den tilltänkta miljön så har kunnat skaffa oss kunskap av den organisatoriska miljön i dessa iterationer. Denna kunskap har sedan applicerats i designprocessen som princip 3, ömsesidigt skapande, förspråkar. Enligt princip 4, ömsesidigt inflytelserika roller, har vi med ett nära samarbete tillsammans med informant 1 utvecklat och evaluerat artefakten där vi främst bidragit med den bakomliggande teorin och informant 1 bidragit med kunskap om organisationen och användarna. Vi har kontinuerligt evaluerat våra alfaversioner på ett formativt sätt där vi tittat på både väntade och oväntade konsekvenser som designen gett. Vi har i artefaktens betastadie evaluerat den på ett summativt sätt där vi har tittat på värdet för dels systemägaren och dels för användarna. Allt i enighet med princip 5, autentisk och samverkande evaluering.

3.4.3 Fas 3, reflektion och lärande

Fas 3 användes genomgående parallellt med de andra faserna under projektets gång, detta för att vi inte skulle ha ett allt för specifikt fokus. Dessutom gav det oss en reflektiv ansats i och med att vi alltid tänkte på hur, varför och vilka aspekter som påverkade vår design av lösningen. Vi tar upp de lärdomar vi kommit över efter varje iteration i fas 2 för att sedan sammanfatta dessa i avsnitt 4.3.

3.4.4 Fas 4, formalisering av lärande

I fas 4 har vi anammat princip 7, generaliserat resultat. Här har vi tittat på vår lösning och generaliserat den till den klass av problem vi definierat i fas 1. Detta avhandlas i vår diskussionsdel avsnitt 5.

3.5 Datainsamling

I detta avsnitt går vi igenom hur arbetets datainsamling utförts.

3.5.1 Intervjuer

För att få svar på vår frågeställning har vi valt att göra en kvalitativ fallstudie där vi använt oss av semistrukturerade intervjuer. Vi har valt att intervjua systemansvarig samt två slutanvändare. Genom att intervjua parter på båda sidorna (de som ger feedback och de som tar emot den) så får vi en bred bild på hur situationen ser ut hos fallstudieobjektet. Intervjuerna i kombination med tidigare litteratur utgör grunden till skapandet av den artefakt vi skapat som sedan generaliseras till den klass av problem vi tittat på.

Intervjuerna spelades in där vi sedan kunde transkribera och citera dem i vårt arbete. Vi informerade våra informanter om varför vi gjorde intervjuerna och vad de skulle användas till innan och frågade om lov om att spela in samtalen. Med informant 1 skedde detta muntligt men informant 2 och 3 informerades via e-post (se bilaga 6). Vi har beskrivit detaljer kring intervjuerna i tabell 2.

3.5.2 Intervju med informant 1

Den inledande intervjun med informant 1 i rollen som utvecklare och informant strukturerades upp efter teman, för att styra samtalet i de riktningar vi ville men med gott om utrymme för följdfrågor. I enlighet med Patel och Davidsons (2011) resonemang om standardisering och strukturering så valde vi att ha en låg grad av dessa då vi ville göra en kvalitativ analys av resultaten. Denna intervjus syfte var att fånga problembilden så att vi kunde göra en adekvat problemformulering. I och med intervjuns öppenhet så framkom även lösningsförslag i denna intervju som sedan används som grund till första iterationen i andra fasen.

3.5.3 Intervjuer med slutanvändare

Syftet med intervjuerna var att hitta slutanvändare som kan vara med och utföra den summativa evalueringen. På grund av att vi inte hade möjlighet att träffa några slutanvändare så har dessa intervjuer skett via telefon. Vi har innan intervjuerna skickat de frågor vi har till informanterna så att de kan förbereda sig på ett så bra vis som möjligt (se bilaga 2). Intervjuerna har haft en hög grad av standardisering då vi har valt att ställa mer konkreta frågor på grund av att vi visste bättre vad det var vi ville ha ut av intervjuerna än vad som var fallet med informant 1. Struktureringen av intervjuerna var låg då vi ville ge informanterna utrymme att svara på våra frågor och oss utrymme att ställa följdfrågor (Patel & Davidson 2011).

3.5.4 Evaluering

Det är lätt hänt att designers och användare gör olika antaganden kring terminologi, möjliga svar på formulär och andra frågor i användningssammanhang. Därför menar Tidwell (2010) att även om man är nästan säker på att designen är bra så ska man göra användbarhetstester. Dessa ger då empiriska bevis på vad som fungerar och vad som inte fungerar.

I alfastadiet av vår artefakt har evalueringen i iterationerna tillsammans med informant 1 varit formativ. Enligt Sharp et al. (2011) görs formativ evaluering under tiden man designar för att se om artefakten fortsätter att följa och stödja användarnas behov. När vi sedan tog fram betaversionen gjorde vi en summativ evaluering tillsammans med informant 1,2 och 3. Summativ evaluering är en bedömning hur lyckad den slutgiltiga produkten är gentemot de krav man utgått från (Sharp et al. 2011).

Fas 1, problemformulering			
Syfte	Deltagare	Tid	Plats
Undersökande intervju	Informant 1	40 minuter	Konferensrum
Undersökande intervju	Informant 2	8 minuter	Telefonintervju
Undersökande intervju	Informant 3	6 minuter	Telefonintervju
Fas 2, UIE			
Syfte	Deltagare	Tid	Plats
Formativ evaluering 1	Informant 1	43 minuter	Personalrum
Formativ evaluering 2	Informant 1	44 minuter	Personalrum
Summativ evaluering	Informant 1	41 minuter	Personalrum
Summativ evaluering	Informant 2	19 minuter	Telefonintervju
Summativ evaluering	Informant 3	17 minuter	Telefonintervju

Tabell 2. Intervjudetaljer.

4 Resultat

Vi kommer i detta avsnitt i kronologisk ordning beskriva vår arbetsprocess. Vi presenterar vårt resultat och diskuterar det kortfattat i reflektion och lärande-avsnitten. Den övergripande diskussionen sker i diskussionsavsnittet. Resultatet är uppdelat i de tre första faserna av ADR, *problemformulering, utveckla, ingripa och evaluera* och *reflektion och lärande*. Den fjärde fasen *formalisering av lärdomar* behandlas i diskussionsavsnittet.

4.1 Problemformulering

För att få en god förståelse för problembilden, så har vi dels hämtat empiriska material till problemformuleringen genom intervjuer med utvecklare och användare av webbtjänsten och dels studerat teori i den existerande litteraturen. Dessa två arbetssätt har gått hand i hand i arbetet med att ta fram en tydlig problemformulering och har resulterat i en samling krav som ställs på vår artefakt. Under inhämtning av det empiriska materialet framkom det en mängd lösningsförslag. Dessa har vi samlat under rubriken Lösningsförslag. Dessa förslag har framkommit i intervjuer som då glidit över i diskussionsform mellan oss och informant 1.

4.1.1 Feedback-inhämtningen innan vår inblandning

Innan vårt arbete så togs feedback in via telefon och mail genom person i supporten. Sedan gjordes den informationen om för att hamna i ett ärendehanteringssystem. Något som anses problematiskt är att kunna ställa rätt frågor.

“..när support har pratat med en kund och så ringer de mig, då har jag kanske fått in 40 % av den information som jag behöver. Så det är upp till oss att fråga kunden vad som behövs.” (Informant 1)

“Det vi vill veta är ofta var i systemet användarna är där de har problem. [...] det kan ju också vara att här hade jag velat ha en knapp. (pekar på bordet) ” (Informant 1)

“..ofta handlar det bara om små funktioner i systemet. Det handlar ju inte om något större, vidare egentligen. Utan det är egentligen bara för att göra systemet smidigare, mer än att utöka det mer större funktioner.” (Informant 1)

Den feedback som kommer in från deras kunder handlar oftast om små förbättringar och inte om att skapa stora nya funktioner i systemet. Vidare menar informant 1 att de vill veta var användaren är i systemet och specifikt vilken funktion de menar. Att få reda på vilken specifik funktion som avses är något som anses hjälpsamt och väldigt högt värderat av utvecklare (Pagano & Bruegge 2013; Lohmann & Rashid, 2008).

Lohman och Rashid (2008) nämner två olika dimensioner för fjärrdeltagande, självstyrande eller händelsedrivna. Självstyrande deltagande är när användaren rapporterar feedback spontant när den

känner för det inne i systemet. Händelsedrivna feedback är när systemansvariga uttryckligen inbjuder användarna till att ge feedback vid en speciell tid. Men vår informant motsätter sig det senare angreppssättet med citaten:

“Det finns en nackdel i och med att vi säger att vi tar emot feedback. För då sekunden efter så säger de ”Jo men jag har tänkt på det här”. Sen att det är en helt oväsentlig sak. Men personen blev precis tillsagd att den ska ge feedback, då gör personen det, fast den inte har något att säga.” (Informant 1)

“Innan hade vi att man kunde lämna feedback, problemet var då att de lämnade en massa feedback och vi kunde inte svara på all feedback, hantera den och ge bra konstruktiva svar. Kan de bara få tillbaka att vi satt en prio på den så ser de det. Så räcker det, vi behöver inte skriva något mer, vi gör något för vår del och användarna får se det.” (Informant 1)

Tidigare har de haft en liknande enkel feedback-funktion där man skriver in rubrik och en kommentar. Men att hantera den och ge konstruktiva svar var svårt eftersom feedbacken kan vara svår att tolka när man inte är klar över vilken specifik funktion de syftar på.

4.1.2 Krav

Problem som framkommit i den empiriska undersökningen är i grunden att Struktur vill ha bättre strukturering över feedback-processen genom en bättre form av inhämtning av feedback från användarna och en ökad transparens gentemot användarna i hanteringen av dessa ärenden. Utifrån Lohmann och Rashids (2008) nyckelaspekter för att minska barriären för användarna att delta (se avsnitt 2.2.2), och från den specifika situationen som vi undersöker har vi kommit fram till följande krav.

Krav	Förklaring
Enkelhet och assistans	Funktionerna ska vara lätta att förstå och väl beskrivna
Lättviktigt deltagande	Funktionerna ska inte hindra användarens nuvarande aktivitet i systemet
Positionering	Användaren ska snabbt och enkelt kunna beskriva var och vad i systemet som feedbacken berör
Transparens	Användaren ska se att den feedback man skickat in hanteras
Inbäddat i systemet	Feedback-funktionen ska vara i systemet, man ska inte behöva gå ur för att lämna feedback

Tabell 3. Tabell över krav utifrån problemformulering.

4.1.3 Lösningsförslag

När kraven (se tabell 3) sammanställts och strukturerats påbörjades en diskussion kring lösningsförslag kring inhämtning av feedback.

Eftersom ett krav var att artefakten ska vara *Inbäddad i systemet* behövde användaren en klar ingång och åtkomst till artefakten, därför användes Tidwells (2011) designmönster (se avsnitt 2.3 för designmönster) *Clear Entry Point*. Ett förslag för det som informant 1 kom med var att alltid ha funktionen längst upp till höger i systemet. Vidare diskuterades hur kravet *Positionering* skulle lösas på ett för användaren acceptabelt sätt. Det som vi valde att utgå från var att användaren ska dra en ikon som symboliserar feedback till sitt problem, hädanefter kallat "drag and drop"-funktionen. På så sätt låter vi användaren enkelt välja var problemet är utan att användaren behöver beskriva det med text och mottagaren av feedbacken får ett sammanhang. Runt kravet *Transparens* diskuterades olika förslag. Ett förslag var att i systemet skulle utvecklarna publicera vad de arbetade på för tillfället, där användarna skulle kunna kommentera arbetet med förslag och kommentarer. Ett annat förslag var att användaren skulle få en återkoppling med vilken prioritering feedbacken fått i Struqturs ärendehanteringssystem. Kravet *Enkelhet och assistans* ska finnas genomgående i arbetet.

4.1.4 Slutsats

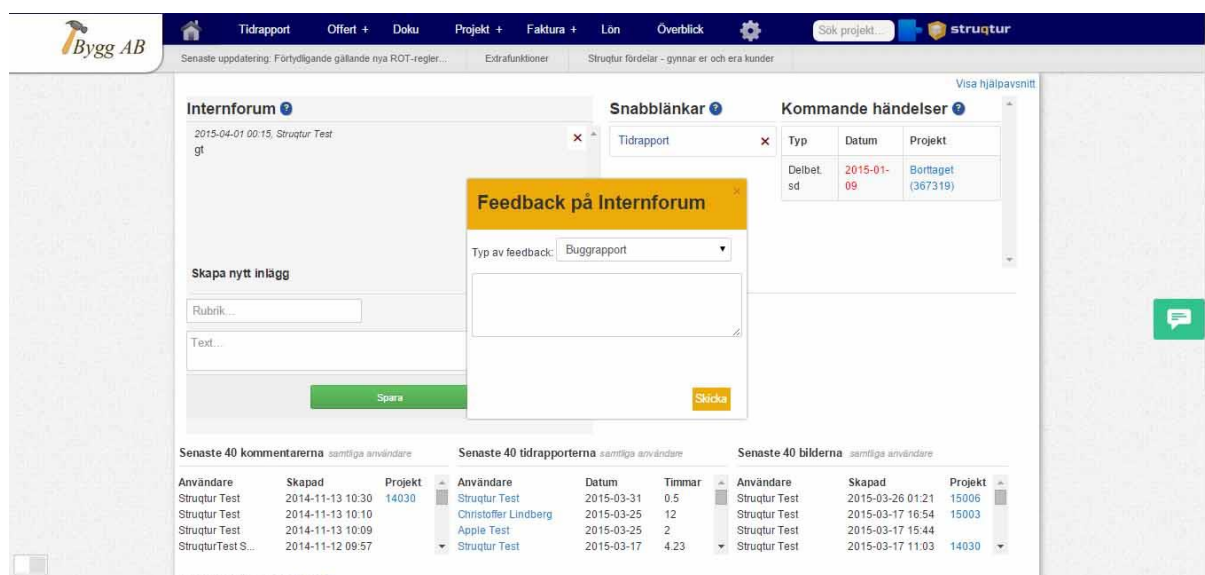
Denna fas gav oss en bra bild av hur det ser ut i nuläget hos vårt studieobjekt i deras inhämtning och hantering av feedback. Utifrån informationen från datainsamlingen i denna fas så har vi formulerat några krav/principer som vår artefakt ska innehålla för att kunna stävja problembilden. Vi definierar klass av problem som användarinvolvering i SaaS-system.

4.2 Utveckla, ingripa och evaluera

Fas 2 utveckla, ingripa och evaluera har utförts i tre iterationer, där vi har utgått från de krav som sammanställts tidigare i problemformuleringen. För att få en så verklig bild av situationen har utvecklingen skett i en testmiljö som är lik det befintliga systemet. Artefakten som skapats har utvecklats som en fristående del som enkelt ska kunna sättas in eller plockas ut ur systemet, den påverkar inte övrig funktionalitet för användarna. Att utveckla på så sätt, i komponenter, är något som rekommenderas av Papazoglou och Ribbers (2006) på grund av att det minskar underhållningskostnaderna. Eftersom artefakten är inkapslad kan man göra förändringar med den utan att påverka de andra systemen som använder sig av den.

4.2.1 Iteration 1

I första iterationen utvecklades en väldigt enkel alfaversion för att se om det finns någon faktisk nytta för Struqtur samt hur lätt den upplevs att använda. Funktionaliteten i alfaversionen var väldigt begränsad och det lades inget fokus på utseendet. Alfaversionen utvecklades utifrån två krav i problemformuleringen att det skulle vara enkelt för användaren att lämna feedback och att den ska samla in vilken del i systemet som användarna åsyftar, detta för att underlätta för support och utvecklare av systemet.



Figur 3. Skärmdump av artefakt iteration 1.

I denna iteration användes designmönstret *Clear Entry Point*. Det finns bara ett sätt att använda sig av feedback-funktionen och det är via ikonen till höger mitt på sidan (se figur 3) som är i en annan färg än övriga systemet för att utmärka sig. Ikonen används med "drag and drop". När användaren drar ikonen över en sektion så blir den sektionens bakgrund grå. Detta för att användaren ska förstå vilken sektion på sidan som åsyftas när den släpper ikonen. När användaren släpper ikonen så kommer det upp ett fönster för inmatning av information.

I formuläret så ställs användaren inför att välja vilken typ av feedback det är. För att göra det enkelt för användaren samt lättare med kategorisering av feedbacken så används en combo box med förvalda kategorier (buggrapport, förslag, annat), samt ett textfält där användaren beskriver sin feedback. Det som sedan samlas in från användaren när den har fyllt i och skickat sin feedback är vilken användare det är, vilken sida, vilken sektion på sidan, tid, kategori av feedback samt användarens egen kommentar.

Evaluering

Utvärderingen i iteration 1 utfördes genom en kombination av en enkel observation och intervju med informant 1 som sedan följdes av en diskussion om tänkta lösningsförslag. Vi ville främst höra om detta var något att gå vidare på och vad informanten ansåg om "drag and drop"-funktionen.

"Det där att dra är jättebra och att man får det på "papper" så att man ange information direkt. För ett problem var ju också att det ska vara så fördelaktigt för att användarna att mata in information, men för oss ska det också bli intressant att ta hand om den. Med detta upplägg så finns det goda möjligheter att göra det" (Informant 1)

Enkelheten för användaren att välja var i systemet de har problem med att dra ikonerna istället för att försöka beskriva det med ord uppskattas av informant 1, dessutom så tror informanten att den information som samlas in är till nytta för dem att lägga in i deras ärendehanteringssystem.

“Om det här fungerar får vi ner mängden av telefonsamtal. Jag tror också att användarna måste få en siffra, antal svar de har fått på sin feedback. De lämnar feedback, när supporten går igenom och placerar dem och skriver en kommentar. Då ska de få upp att något har hänt med ärendet, för att uppmuntra feedback.” (Informant 1)

I och med enkelheten så tror informant 1 att användare kan föredra detta sätt att komma med synpunkter och att det kommer underlätta för deras support. Men att användare måste få något tillbaka av för att fortsätta med inlämning av feedback.

Slutsats

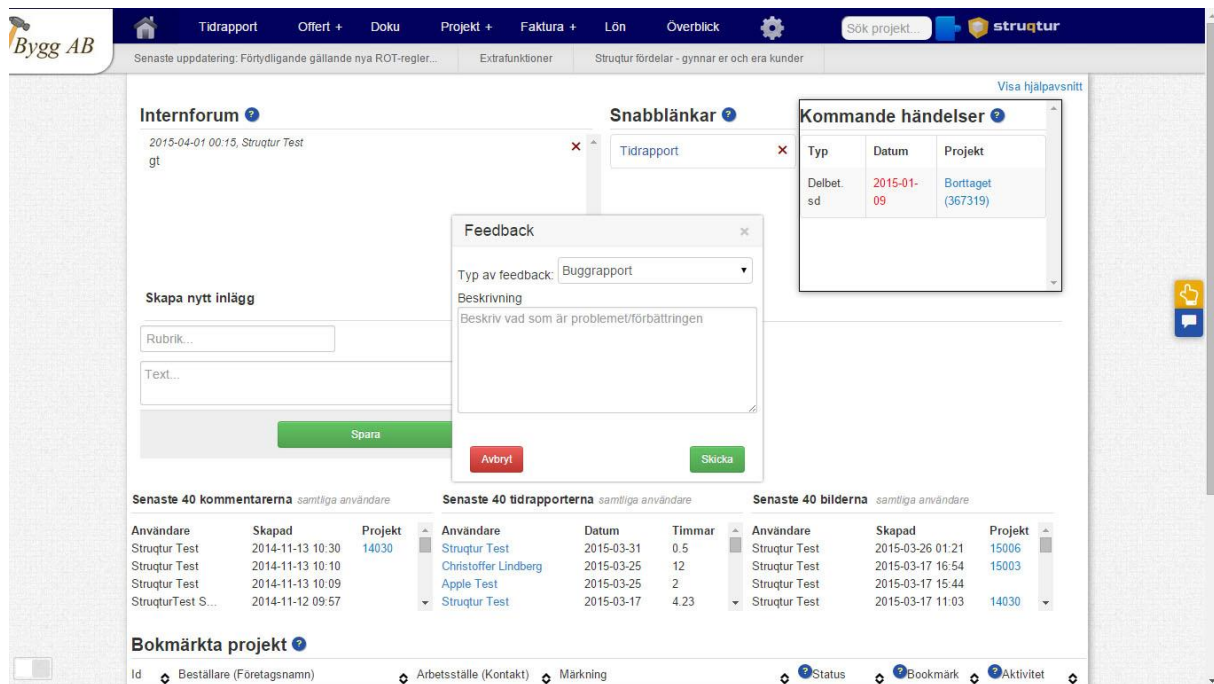
Den första iterationen var kort och utfördes snabbt eftersom vi ville utvärdera om det här skulle vara något för systemet Struqtur. Alfaversionen mottogs positivt och de ser en nytta för deras organisation och användare. Med denna komponent kan de hämta in supportärenden som tidigare kommit in via telefon och lagts till manuellt i deras ärendesystem. Det vi tog med oss från denna iteration var en förstärkning av vårt krav transparens. Sedan mottogs ”drag and drop”-funktionen som ett bra sätt att länka feedback till en del av systemet.

Reflektion och lärande

Syftet med artefaktens utseende i iteration 1 var i huvudsak att utröna hur ”drag and drop”-funktionen upplevdes i systemet. Ett tydligt problem inom området är att få användaren att beskriva var i systemet man syftar på när de lämnar feedback. Detta stöds både av våra empiriska resultat och den teoretiska datainsamlingen. Att användaren får på något vis visa med en markör var i systemet som kommentaren avser istället för att beskriva detta med text, tyckte vi tillsammans med informant 1 var en mycket bra idé. Att ta tag i någonting, i detta fall en ikon, och släppa den där i systemet man vill påpeka något kändes som en naturlig lösning både för oss och för informant 1. För att öka transparensen mot användarna satsar vi på något som ska visa användaren att feedbacken mottagits och behandlats på något vis.

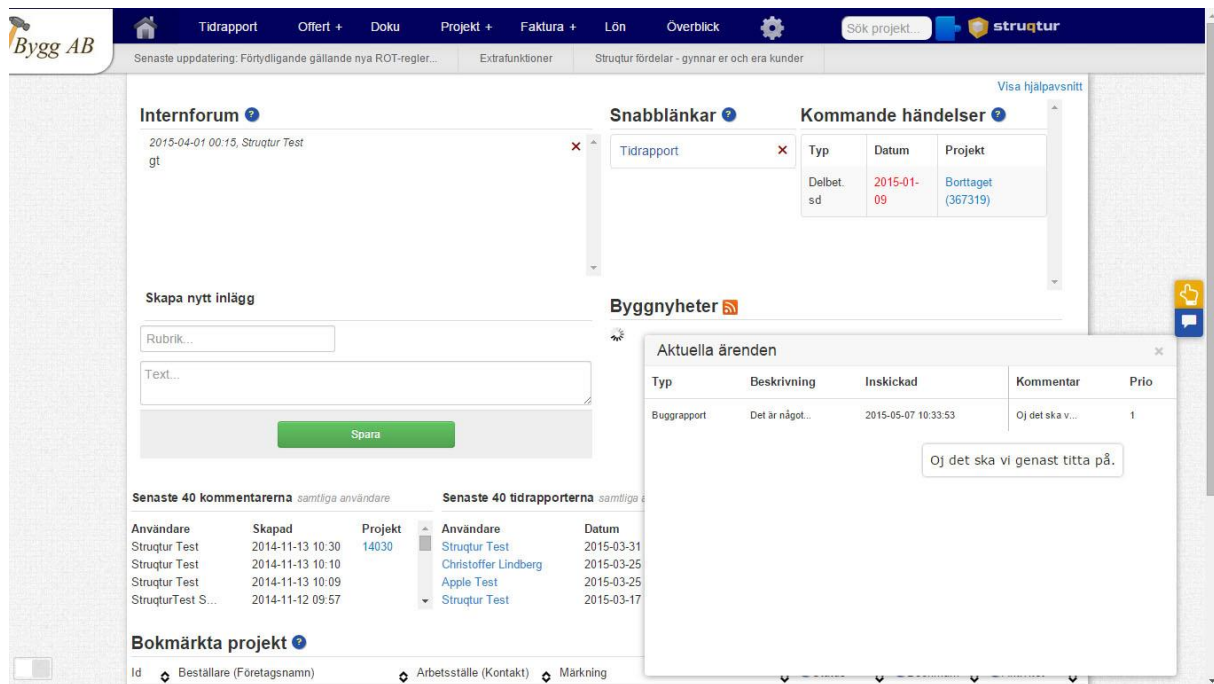
4.2.2 Iteration 2

För att bygga vidare på vår design så var iteration 2 utförligare än den tidigare. Mer vikt lades på utseendet och att ta reda på hur det är enklast för användarna att delge den information som utvecklarna behöver för att förstå problemet/förbättringen. Här använde vi oss av Tidwells (2011) designmönster *Escape Hatch*, *Modal Panel*, *Movable Panel*, *Collapsible Panel* (Se avsnitt 2.3) för navigering och organisering av information.



Figur 4. Skärmdump från iteration 2.

Dessutom lades större vikt vid att det visuella skulle vara mer likt systemet som artefakten ska implementeras i. Färgschemat anpassades efter vilka färger som redan nu finns i systemet. Designmönstret *Modal Panel* används när man släpper ikonen, för att hindra att användaren gör något annat i systemet när den fyller i formuläret. *Escape Hatch* används för att användaren klart ska förstå hur man kan gå ur feedback-läget. Detta görs med en röd knapp nere till vänster på formuläret och ett grått kryss uppe till höger på formuläret. För att inte hämma användarna att se vad för element på sidan som markerats och var så är panelen flyttbar enligt mönstret *Movable Panel*.



Figur 5. Skärmdump iteration 2.

För att hjälpa användarna förstå "drag and drop"-funktionen så lade vi en handsymbol i ikonen för att symbolisera att man ska dra den och muspekaren ersattes med ett kors med pilar när muspekaren var över ikonen. Dessutom lade vi dit en liten textruta med texten "Dra handen och släpp där du vill lämna feedback" som visades när muspekaren var över ikonen.

När användaren släppt handikonen någonstans i systemet så markeras detta genom en svart ram runt objektet (se *kommande händelser* i figur 4). Vi kände att det var viktigt att få denna feedback från systemet så användaren ser var artefakten registrerade att användaren släppte ikonen.

Designmönstret *Collapsible Panel* används när man klickar på den blå ikonen som visar en pratbubbla. När den klickas på så kommer det ut en panel där man kan se sina aktuella ärenden (figur 5) som innehåller en kommentar. Detta görs för att kunna få in en stor del information som inte behöver ta någon plats om man inte aktivt väljer att titta på den. Valet att man ska kunna se sina lämnade ärenden hör ihop med vårt krav på transparens.

Evaluering

Fyra huvudsakliga beröringspunkter kom upp under evalueringen av iteration 2: 1) det är för otydligt vad ikonerna syftar till, 2) ikonerna tar för mycket plats och fokus, 3) *Aktuella ärenden* är inte tillräckligt med information om behandlade ärenden och 4) det går inte att beskriva ett allmänt problem.

Otydligheterna i ikonerna var ett problem då vi inte förtydligat på något vis att de avser just feedback. Handen kan vara en bra symbol för att visa att det är tänkt att man ska ta och dra den men då behöver man sätta den i ett sammanhang, ge användaren ett syfte att ta och dra den. Den textruta som dök upp när musen hölls över var inte tillräckligt informativ i detta avseende.

Att ikonerna tog för mycket plats och fokus i förhållande till hur ofta de kommer att användas var också någonting som behövdes ses över. Vi diskuterade möjligheten att lägga en mindre knapp med tydlig "feedback-markering" nere i högra hörnet istället. Det finns en verktygsknapp nere till vänster i systemet. Detta gjorde det rimligt att lägga vår knapp nere till höger av två anledningar: 1) den symmetriska aspekten och 2) att det finns mer plats på högersidan än vänstersidan då det inte fanns några knappar där sedan tidigare.

Angående hanteringen av *Aktuella ärenden* så insåg informant 1, när hen nu för första gången sett en lista på aktuella ärenden för användaren, att det behövdes mer än bara en visning av de ärenden som kommenterats och prioriterats. Att användaren även ska kunna se avslutade ärenden kändes nu självklart. Vi diskuterade hur man skulle lösa att notifiera användaren om att ett ärende blivit flyttat från *Aktuella ärenden* till *Avslutade ärenden*. Notifieringen kändes viktig på grund av att Struqtur inte vill förvillan användaren om vad som hänt när ett ärende från *Aktuella ärenden* försvinner då det blivit löst. Vi bollade lite idéer om att använda samma typ av notifiering som när ett ärende blir behandlat och inlagt i *Aktuella ärenden*.

En sak som helt hade förbisetts var om användaren vill beskriva ett allmänt problem eller en generell förbättring, till exempel om systemet går segt. Detta är ju då inte någonting som man kan peka på en speciell position i systemet. På något vis behöver man också kunna ge feedback som inte är förknippas med en speciell position eller vy.

Det dök även upp en fråga från informant 1 om ärendena visades på användarnivå eller på företagsnivå. Vi hade hela tiden utgått från användarnivå, men en god anledning till att lägga visningen på företagsnivå istället var att användaren då kan se om någon annan på företaget redan rapporterat in det just det användaren vill rapportera in. På så vis skulle Struqtur kunna slippa "dubbletter". Dessutom så menade informant 1 att det skulle bli mindre personligt vilket hen förmodade att användarna skulle motta positivt. Motargumentet vi kunde hitta för företagsnivå var att Struqtur kanske missar data om hur många som upplever problemet eller har ett specifikt önskemål. Det framkom dessutom önskemål om att användaren själv ska kunna sortera listorna med ärenden efter alla de olika attributen i listan med undantag för *Kommentar* och *Beskrivning*.

Slutsats

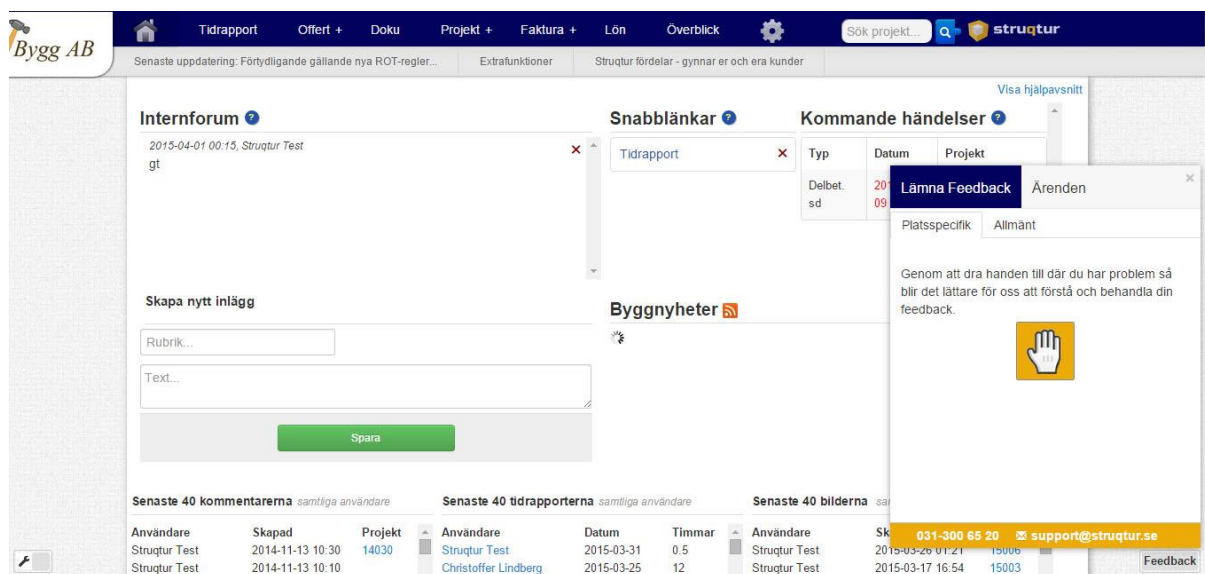
Iteration 2 handlade om att få in den funktionalitet i artefakten som ansågs nödvändig men inte mer då den skulle hållas så enkel som möjligt för att underlätta för användarna. Viktigt var också att förverkliga de idéer som dök upp i problemformuleringen och i iteration 1 så att vi tillsammans med informant 1 kunde ta ställning till någonting konkret istället för abstrakta idéer.

Reflektion och lärande

Att hålla någonting så avskalat och enkelt som möjligt men att det ändå ska ha tillräckligt funktionsinnehåll, i vårt fall en feedback-funktion, är en svår balans. Efter denna iteration stod klart att vi behövde öka funktionaliteten för att faktiskt öka användarvänligheten. Vi hade lagt oss på en för låg nivå och på så vis gjort det svårare för användarna. Något som däremot upplevdes som en väldigt bra funktion, även om den behövde omdesignas lite, var *Aktuella ärenden*. Det ger användaren en god insyn i systemägarnas arbetsgång med det specifika ärendet vilket uppfattas som att användaren blir sedd och hörd.

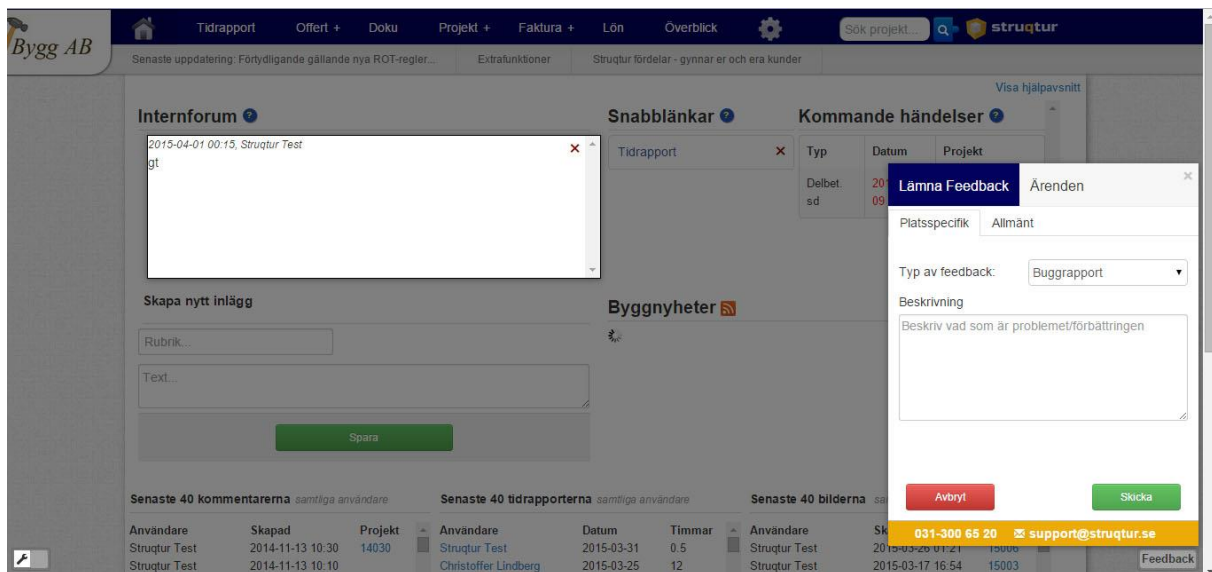
4.2.3 Iteration 3

Utvecklingen i iteration 3 resulterade i att sättet användaren kommer åt feedback-funktionen ändrades. Vi följde de önskemål som kom upp om detta i evalueringen av iteration 2. Istället för två separata ikoner som tillgängliggör funktionerna så lades en knapp nere till höger med texten "Feedback" som kan ses i figur 6 längst ner till höger.



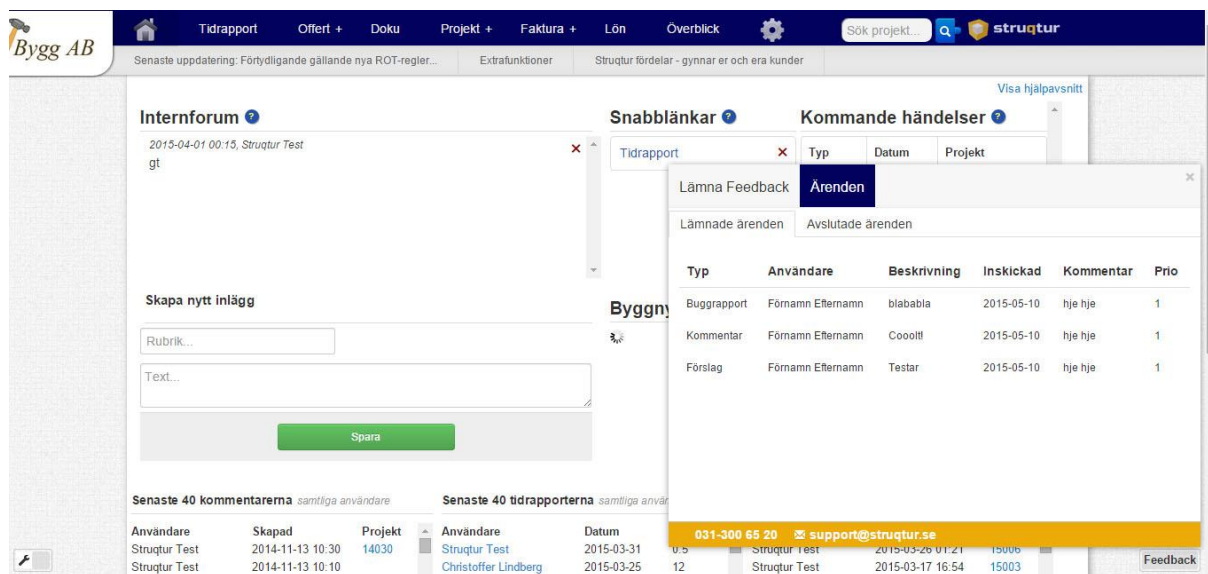
Figur 6. Skärmdump på "Drag and drop"-funktionen i iteration 3.

Anledningarna till att vi valde att lägga all funktionalitet i en statisk ruta med flikar (se figur 6) som kommer upp ur högra hörnet, istället för att använda flera knappar till de olika funktionerna som tidigare var flera. För det första så utökades antalet funktioner från handikonen och *Aktuella ärenden* till två olika val under *Lämna Feedback*, *Platsspecifik* och *Allmän*. Och två val under *Ärenden*, *Lämnade ärenden* och *Avslutade ärenden*. Med så många val så kände vi att vi behövde gå ner nivåmässigt och inte ha en knapp för varje funktion i högsta nivå. Därför designades flikarna enligt Tidwells (2011) designmönster *Module Tabs*, som används när man vill få mycket information på liten yta som inte behöver synas på samma gång. En annan anledning till förändringen var tydligheten, att få in text kändes nödvändigt då vi inte lyckats hitta självklara ikoner.

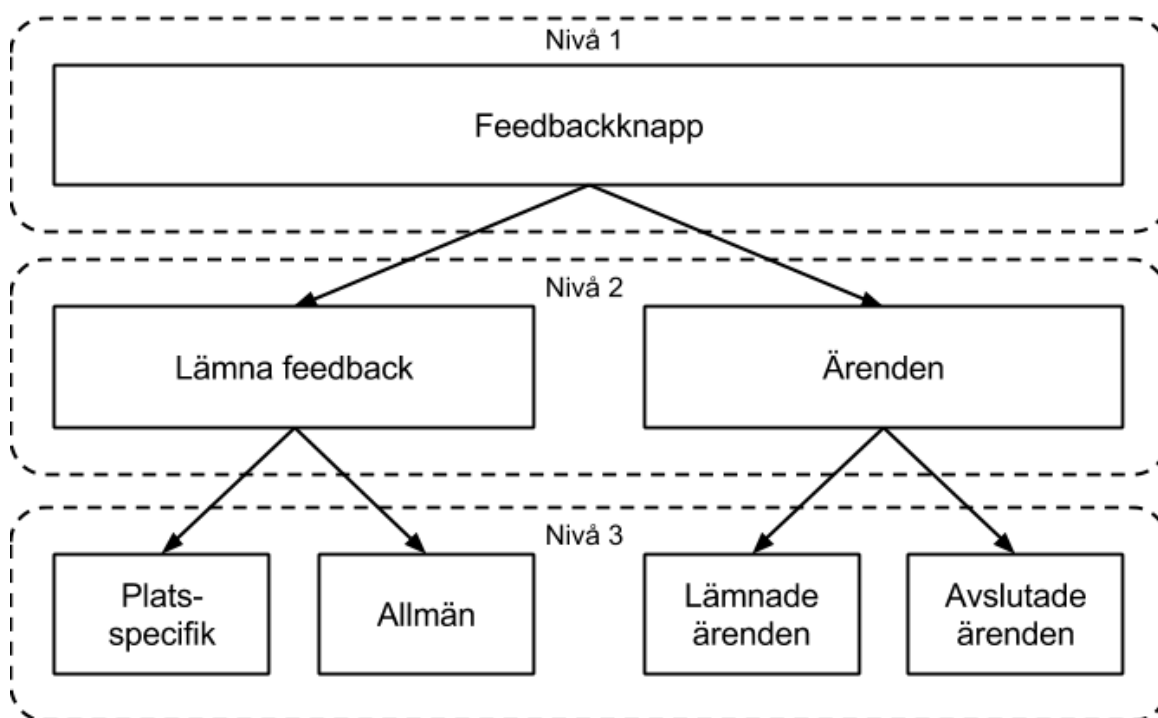


Figur 7. Skärmdump från Iteration 3.

Funktionen *Lämna Feedback* delade vi in i två kategorier av feedback, *Platsspecifik* och *Allmän*. *Platsspecifik* har samma funktionalitet som huvudfunktionen i iteration 2, det vill säga "drag and drop". Denna funktion är det användaren möts av vid klick på *Feedback-knappen* (se figur 6). Om användaren vill lämna feedback som inte går att knyta ihop med en specifik plats i systemet, till exempel att systemet är segt, så får hen klicka på fliken *Allmän*. Vid lämnning av allmän feedback så dyker textinmatningsrutan upp direkt och steget med "drag and drop" används inte. En skillnad från iteration 2 är att inmatningsrutan inte hamnar mitt på skärmen när man släppt handen på ett objekt i systemet utan dyker upp i den rutan där man hämtade handen ifrån (se figur 7). Detta hänger ihop med att när användaren lämnar allmän feedback var det naturligt att lägga textinmatningen i feedback-rutan. Därför ville vi göra samma sak när användaren lämnade platsspecifik. För att göra det tydligare var användaren släppt handsymbolen så drog vi uppmärksamhet till detta genom att behålla ramen runt objektet och belyste det ytterligare genom att mörka bakgrunden. Detta gjorde vi också för att dra ögonen till inmatningsrutan då denna inte uppstod mitt på skärmen utan ligger inbäddad i feedback-rutan. Vi lade också en fördröjning på uppdateringen av feedback-rutan när textinmatningen kom fram men 750 millisekunder för att ytterligare dra uppmärksamhet till att det händer något nytt med den rutan.



Figur 8. Skärmdump från Iteration 3.



Figur 9. Schema över nivåerna i artefakten.

När användaren lämnat feedback på någonting och systemadministratörerna har kommenterat och gett ärendet en prioritering så används samma notifikationssymbol som tidigare i iteration 2, fast nu vid feedback-knappen. Ett ärende som användaren inte redan sett markeras med en grön bakgrund som sedan försvinner när det nya ärendet är sett av användaren. Fliken *Ärenden* är i sin tur uppdelad i två flikar, *Lämnade ärenden* och *Avslutade ärenden* (se figur 8 och 9). Ett tillägg bland attributen i tabellen med ärenden är *Användare*. Detta attribut tillkom då ärendena presenteras på företagsnivå istället för användarnivå. I dessa två flikar ser struktureringen lika ut. Enda skillnaden är statusen på

ärendet, om det är avslutat eller om det endast är kommenterat och har blivit tilldelat en prioritering.

Summativ evaluering

Den summativa och slutgiltiga evalueringen i iterationsprocessen genomfördes med alla informanter i intervjuform. Evalueringen med informant 1 skedde i samma format som de tidigare evalueringarna. Vi gav informant 2 och 3 tre uppgifter att lösa med hjälp av artefakten (se bilaga 3) och sedan intervjuade vi dem. Nedan följer en sammanställning av evalueringen med alla tre informanter.

Överlag var alla väldigt nöjda med artefakten. Det som drog mest uppmärksamhet var ”drag and drop”-funktionen och att användaren kan se de ärenden som kommenterats, prioriterats och slutligen avslutats. Reaktionen på ”drag and drop”:

”Väldigt enkelt, jag gjorde som jag blev tillsagd att göra. Jag gillar det med handen där, det var bra att man kunde dra i den så slipper man förklara så mycket.”(informant 2)

”Man tar ju bara handen och drar dit det krånglar och så ser de (Struktur) ju var det är” (informant 2)

”Jättelätt, om man nu ska rapportera Byggnyheter så är det ju bara att dra dit handen, och sen beskriva typ och med text” (informant 3)

”Den är mycket mer logisk när den kommer upp så här, (jämfört med tidigare versioner).” (informant 1)

”Drag and drop”-funktionen har varit en stor del i arbetet med artefakten. Problemet med att beskriva var i systemet användaren avser när denne lämnar feedback har varit tydligt genom hela processen. Här får vi kvitto på det upplevs positivt att beskriva det grafiskt istället för med text eller via tal upplevs som någonting väldigt positivt hos användarna.

När de lämnat feedback så noterade användarna att det kommit in ett nytt ärende. Att de ges möjlighet att ta del av ärendehanteringens upplevdes som någonting positivt och enligt informant 3 som nödvändigt. Reaktionen på det:

”Jag tyckte det var jättebra, då vet man att det är registrerat. Annars går jag bara och undrar så det gillar jag” (informant 2)

”Mycket bra att man får ett svar. Det krävs ju, om man säger hej i telefonen så vill man ju att den andre ska säga hej tillbaka annars skiter man ju i att ringa nästa gång” (informant 3)

Även från organisationens sida så såg man fördelar:

"Om användaren då ser att det inte är prio för oss... Man sparar mycket tid på den kommunikationen." (informant 1)

Något som visade sig vara lite problematiskt med vår artefakt var att båda användarna upplevde att den var svår att hitta och att den var för undgängd i systemet. Informant 2 uttryckte det på följande vis:

"jag upptäckte den inte först, det hade inte gjort något om den tog lite mer plats"
(informant 2)

Informant 3 stämmer in i det uttalandet och ger sin syn på hur den borde se ut och vara placerad:

"Den var svår att hitta, det hade varit bättre att den låg där man rör sig med musen ofta och en stor färgglad knapp... Det är lätt annars att man glömmer att det finns en feedback-funktion" (informant 3)

Detta är intressant då informant 1 i rollen som utvecklare uttryckligen tyckte att den tidigare tog för stor plats och att detta var betydligt bättre än iteration 2.

När vi frågar våra informanter om helhetsintrycket av artefakten så får vi ett positivt gensvar:

"Vi gillar den, det är lättare att kommunicera känner jag" (informant 2)

"Jag visade den för min kollega också, den var suverän för oss. Enkelt och snabbt och man kan få svar snabbt" (informant 2)

"Mycket bra, det blir ju lättare för oss att visa var vi menar och lättare för dem (Struktur) att kunna hantera ärendet snabbare" (informant 3)

"Jag tror jag hellre skulle använda den här funktionen än att ringa... åtminstone när det är mindre saker som det gäller, blir det väldigt komplicerat ringer jag nog hellre"
(informant 3)

"För den här kan vi lägga in i allting vi byggt, alla är så trötta på e-support att folk ringer och mailar om problem." (Informant 1)

Detta är något som Struktur skulle kunna lägga in i sitt system, de ser en stor nytta för organisationen i och med att det skulle få ner mängden telefonsamtal till deras support, även användarna skulle kunna använda detta istället för att ringa.

4.2.4 Slutsats

De krav vi ställde på artefakten i problemformuleringen (se tabell 3) har, tillsammans med de designmönster vi valt att jobba med (se avsnitt 2.3), väglett oss under utvecklingsarbetet. Dessa har fallit ut väl som beskrivet i den summativa evalueringen. De saker som sticker ut gentemot mer

traditionella feedback-funktioner är ”drag and drop”-funktionen och transparensen i ärendehantering. De har också varit de som mottagits med de största positiva reaktionerna.

För att lösa kravet *Enkelhet och assistans* har vi jobbat mycket med designmönstret *Clear Entry Point*, att beskriva funktionerna på ett så tydligt sätt som möjligt. Ett bra exempel på något som föll väl ut var handikonen som alla som testade instinktivt förstod vad den antydde. Vi har också jobbat textförklaringar som dyker upp när användaren sätter musen över objekt i artefakten så kallade Tool Tips (Tidwell 2011). Att feedback-funktionen upplevdes som svår att hitta får ses som misslyckande. Även designmönstret *Modal Panel* har varit till nytta för detta krav. Eftersom det begränsar valmöjligheterna för användaren, de behöver bara fokusera på en sak.

Det andra kravet *Lättviktigt deltagande* löste vi med hjälp av designmönstret *Escape Hatch*. På så vis kan användaren snabbt och enkelt återgå till att använda systemet.

En utmaning vi stod inför var att hitta ett sätt för användaren att visa var i systemet feedbacken avser, därav kravet *Positionering*. Vår ”drag and drop”-funktion har fungerat mycket väl i denna miljö. Att användaren får beskriva det grafiskt och inte i text ser vi som en stor vinning.

Kravet *Transparens* grundar sig i att förbättra kommunikationen med användarna, något som vi lyckades bra med. Citaten från informant 1 bekräftar både vikten av det och vår framgång.

Vendium AB har som intention att integrera vår artefakt i Struqtur. Detta kommer att ske efter färdigställningen av detta arbete så vi har tyvärr inte kunnat dokumentera det.

4.3 Reflektion och lärande

Vi har genom hela utvecklingsprocessen försökt hålla ner komplexiteten i vår artefakt för att inte skrämma iväg användarna. I den första betaversionen (iteration 2) höll vi ner den så mycket att det blev på bekostnad av funktionaliteten. Detta åtgärdades i iteration 3 och vi kom upp på en funktionalitetsnivå som låg inom de gränser som satts av den rådande situationen. Efter den summativa evalueringen så såg vi inga indikationer på att artefakten var för komplex för användarna. Vi hade hittat balansen mellan mängden funktionalitet och enkelhet.

En intressant situation som uppstod var att när informant 1 i rollen som utvecklare upplevde att artefakten tog lagom med plats i systemet (när den var passiv), så upplevde informant 2 och 3 i sina roller som användare att den tog för lite. Uppenbarligen krävs mer efterforskningar och arbete i med att hitta balansen mellan att synas väl men kanske bli distraherande och vara diskret men riskera att missas när den behövs.

5 Diskussion

Syftet med detta arbete var att undersöka vilka designmönster som är användbara vid utveckling av feedback-funktioner i SaaS-miljöer. Undersökningar har utförts med ADR-metodik och ett feedback-verktyg har utvecklats och evaluerats.

De designmönster som använts i resultatdelen i detta arbete kommer att behandlas var för sig, förutom *Collapsible Panels* och *Module Tabs* eftersom de berör samma ämne. Detta i avseende att finna vilka som kan lyftas fram från vår specifika situation till att generaliseras till att användas i SaaS-system vid design av en feedback-funktion.

Clear Entry Point

Eftersom feedback-inhämtning sällan är huvudfunktion i system så behöver funktionen inte ta stor plats, men ska ständigt finnas tillgänglig för användaren. Detta fann vi problematiskt. I resultatet fann vi att användarna gärna ville att funktionen skulle ta större plats, för att den lättare ska hittas och på sätt snabbt kunna användas när de har problem. Å andra sidan så ville utvecklaren att den skulle ha en liten plats i systemet, för att inte vara i vägen för resterande funktioner. Det gäller att hitta en medelväg. Funktionen ska vara fullt synlig, speciellt för nya användare, men inte hindra visualisering av övrig funktionalitet. Tidwell (2011) menar med att detta är viktigt för nya och icke frekventa användare, då det minskar bördan med att lära sig en ny sida. Designmönstret *Clear Entry Point* får därför anses som viktigt att ha i åtanke när man utformar en feedback-funktion i SaaS-system.

Escape Hatch

För att kunna lösa kravet lättviktigt deltagande (se tabell 3) så behövde vi hitta sätt för användaren att enkelt kunna stänga ner vår artefakt och fortsätta med det vanliga arbetet. Vår artefakt ansågs av användarna som enkel att stänga. De använde sig både av det grå krysset längst upp till höger och av feedback-knappen som de öppnade funktionen med (se figur 8). I och med att vi testat detta i en testmiljö och inte i det befintliga systemet så har vi inte helt klart för oss om vi uppfyllt det kravet på ett bra sätt. Vi anser att *Escape Hatch* är ett viktigt mönster eftersom att en feedback-funktion är en liten del av ett system och lätt ska kunna stängas ner av användaren. Därför kan det appliceras generellt på feedback-funktioner i SaaS-system.

Modal Panel

I iteration 2 (avsnitt 4.2.2) så användes mönstret *Modal Panel* när användaren pekat ut ett element i systemet. Vi upptäckte under utveckling att användarna kan råka skriva eller trycka på knappar i bakgrunden. Detta förhindrades eftersom det skulle kunna leda till problem för användaren. Med detta mönster så kan inte användaren göra något annat än att antingen skriva klart formuläret eller

att stänga ner det. Detta mönster rekommenderas vid utformning för att undvika att problem uppstår när man använder sig av en feedback-funktion i ett SaaS-system.

Movable Panel

I resultatet i iteration 2 (avsnitt 4.2.2) så upptäcktes ett problem när användaren valt ett element i centrum av skärmen, då formuläret som kommer upp hindrar användaren från att se det valda elementet. Användaren kan på så vis lämnas undrande om de verkligen valt ut något specifikt element på skärmen eller inte. För att stävja det problemet användes *Movable Panel*. Om användaren undrar kan den lätt flytta formuläret. Detta togs dock bort i iteration 3, eftersom vi valde att placera formuläret längst ner till höger (se figur 7). Mönstret ansågs hjälpsamt för oss men alldeles för specifikt för att generaliseras till all feedback-inhämtning i SaaS-system.

Module Tabs, Collapsible Panel

Under arbetet så ökade komplexiteten av vår artefakt, och med det så behövdes mer information visualiseras. För att hantera det så användes *Module Tabs* och *Collapsible Panel* (se figur 8) Dessa användes för att artefakten inte skulle behöva uppta en större del av systemet. Information som inte behövs döljs. Vi ansåg att det var viktigt för att kunna lösa kravet på *Transparens* (se tabell 3) som innebar att vi behövde använda oss av mer yta för att infoga mer innehåll. Dessa mönster kan användas vid mer komplexa varianter av feedback men ses inte av oss som primära designmönster för utformning av feedback-inhämtning i SaaS-system.

De fyra nyckelaspekter som Lohmann och Rashid (2008) har tagit fram för att reducera barriären för användaren att delta (avsnitt 2.2.2) kan vi till stor del styrka med våra empiriska resultat. Tre av fyra har visat sig i de reaktioner som vi fått från våra informanter. Vikten av aspekten *Integration i användarens miljö* belyses i våra resultat i form av användarnas önskan om en tydligare markering av funktionen så att den blir lättare att hitta. Artefakten är integrerad i systemet men det var inte tillräckligt tydligt för användarna att funktionen fanns. Användarna trodde att det fanns risk att de skulle glömma att funktionen fanns om inte funktionen visades tydligare. *Lättviktig/enkel delaktighet* har vi inga konkreta indikationer på hur väl vi lyckats med i våra resultat. Vi inte har haft möjlighet att testa artefakten i systemet och på så vis se om användarna upplever deltagandet som lättviktigt. I aspekten *Enkelhet och assistans* belyser Lohmann och Rashid (2008) vikten av att användargränssnittet ska vara enkelt och självförklarande, inte kräva användarna på mycket data eller potentiellt kognitivt krävande klassifikationer samt stödja automatisk formulärsifyllning. I våra resultat har det framkommit med tydlighet att användarna upplever vår artefakt enkel och snabb att använda, vilket i sin tur upplevs som något väldigt positivt. Detta grundar sig i vår ”drag and drop”-lösning. Att positionsbestämma objekt och funktioner grafiskt kan kanske inte räknas till en

helautomatiserad process, men det underlättar väldigt i klassificeringen av feedback och kan inte räknas till en kognitivt krävande process. Vi upplever att denna aspekt har varit en stor del i vår framgång med artefakten. Den sista aspekten *Transparens* har också varit nyttig för oss under utvecklingen. Möjligheten för användarna att se att feedback är mottagen, kommenterad och prioriterad har varit mycket uppskattad hos användarna.

Om man ser till Damodarans (1996) olika former av användarinvolvering (se avsnitt 2.2) hamnar användningen av vår artefakt inom hela spannet. Den är *Informativ* när användarna lämnar feedback kring till exempel buggrapporter. Utvecklaren får hela kontexten med tid, vy, specifik funktion, webbläsare, upplösning, användare samt kommentar. All denna information ger en större förståelse för mottagen feedback. Den är *Konsultativ* när användaren kommenterar på fördefinierade funktioner i systemet, som till exempel fakturahantering eller mindre saker som var en knapp är placerad. *Medverkande* är den när användare kommer på nya förslag som kan påverka systemet i en större grad. Om man ser till Ives och Olsons (1994) grader av inflytande (se avsnitt 2.2) så möjliggör artefakten allt från symbolisk involvering till rådgivande involvering. Symbolisk involvering är det i fall de lämnar in feedback som kanske inte anses som relevant för systemet och läggs därmed in med lägre prioritet i Struqturs ärendehanteringssystem eller kanske rent av tas bort. Rådgivande involvering blir det när tips och åsikter kommer in från användaren som sedan tas hand om av utvecklare för att förbättra systemet.

Ett annat exempel som ansågs nyttigt, men som vi inte funnit som designmönster i litteratur, var att när användaren pekat ut ett element så markerades det med ljus bakgrund och på resten av sidan blev bakgrunden mörk. På så vis har användaren helt klart för sig vilket element den valt.

Vi ser en väldigt stor nytta med att kunna peka var i systemet problemet/förbättringen är positionerat, i vårt fall har detta förverkligats genom en "drag and drop"-funktion. Pagano & Bruegge (2013) påpekar vikten av att förstå vilken funktion som feedback åsyftar. Enligt vad vi erfar efter vårt arbete så är en sådan funktion betydligt enklare för användare att hantera än att beskriva positionen i textform. Däremot är vi inte lika säkra på att just "drag and drop" är det bästa sättet att lösa det på. Funktionen upplevs som smidig och lätthanterlig vid användandet av en traditionell datormus men kan vara krånglig att hantera på till exempel en touch pad eller pekskärm. Det kan argumenteras för att lösa det genom klickning istället, att användaren först klickar på en knapp eller dylikt för att sedan klicka en gång till där i systemet man avser att behandla något, för att på så vis komma undan motoriska bekymmer som "drag and drop" kan medföra. Vi föreslår därför ett nytt designmönster som tillgodoser dessa aspekter.

5.1 Begränsningar

Det faktum vi inte har haft möjligheten att vara närvarande och observera de användare som testat vår artefakt får ses som en begränsning. Det hade varit värdefullt att kunna se hur användarna betedde sig vid användandet och inte bara fått deras subjektiva återberättelse av upplevelsen. Nu var vi beroende av att användaren hade en tydlig minnesbild av situationen. För att undvika att minnesbilden försvagats hos våra informanter så utfördes intervjuerna kort efter evalueringen. Det hade också varit bra med testande i en större skala med fler användare under längre tid i systemet och inte bara i vår testmiljö. Det kan också ses som en begränsning att vi i undersökningen bara arbetat mot ett system, i fall vi även testat detta i andra SaaS-system med sina användare skulle vi med en större grund kunna generalisera vårt resultat.

6 Slutsats

Eftersom vårt syfte var att utforska hur man kan designa en komponent för att inhämta kvalitativ feedback från användare av SaaS-system, för att på så vis förbättra användarinvolveringen så arbetade vi med frågeställningen "Vilka designmönster kan användas för att skapa en miljö i ett SaaS-system som gör att användarna enkelt kan ge kvalitativ feedback?". Det som vi upptäckte under vårt arbete är att designmönstren *Clear Entry Point*, *Escape Hatch*, *Modal Panel* är de designmönster som kan appliceras generellt på SaaS-system för inhämtning av feedback, eftersom de möjliggör ett enkelt och lättviktigt deltagande. Vi upptäckte även under arbetets gång att det inte finns några konventioner varken i litteratur eller i praktik till hur man på ett självförklarande sätt designar en funktion där användaren positionsbestämmer var problem är i systemet. Vår utformning och lösning på problemet var en "drag and drop"-funktion som mottogs positivt av både organisation och användare. Feedback som är positionsbestämd ger en högre kvalitet då mottagaren får en större kontextuell förståelse och kan hantera den bättre. Genom detta förbättras användarinvolveringen.

6.1 Studiens relevans och överförbarhet

Vi har utfört denna studie tillsammans med och på uppdrag av Vendium AB. Vi har utvecklat en feedback-funktion till deras SaaS-system Struqtur i syfte att utröna vilka designmönster som kan vara applicerbara vid utveckling av liknande produkter i SaaS-system. Att vi har begränsat oss till SaaS-system beror på dess känslighet i en användaracceptansaspekt. Studiens relevans grundar sig i denna känslighet och SaaS-systemens utbredning.

Vårt fokus har legat på att lyfta fram generella designmönster som går att applicera på andra system och därför är överförbart. Vi ser heller inga hinder med att överföra denna studie till andra typer av IT-system som har liknande förutsättningar men andra affärs- och/eller leveransmodeller.

6.2 Förslag till vidare forskning

Vi hade inte möjlighet att implementera vår artefakt i systemet och på så vis inte har kunnat göra studier för att undersöka effekter av dess användande. Något som därför skulle det vara intressant att utforska är huruvida en implementering av en feedback-funktion med liknande transparens är något som påverkar användarnas attityd gentemot systemet på längre sikt. Detta skulle kunna göras i en longitudinell studie som jämför användarnas attityd innan det införs med deras attityd efter till exempel ett år införandet.

7 Referenser

Alam, I. (2002). An exploratory investigation of user involvement in new service development.

Journal of the Academy of Marketing Science, 30(3), ss. 250-261.

Armbrust, M., Fox, A., Griffith, R., Joseph, A. D., Katz, R., Konwinski, A., Lee, G., Patterson, D., Rabkin, A., Stoica, I. & Zaharia, M. (2010). A view of cloud computing. *Communications of the ACM*, 53(4), ss. 50-58. DOI:10.1145/1721654.1721672

Barki, H. & Hartwick, J. (1994). Measuring User Participation, User Involvement and User Attitude. *MIS Quarterly*, 18(1), ss. 59-82.

Benlian, A. (2011). Is traditional, open-source, or on-demand first choice[quest] Developing an AHP-based framework for the comparison of different software models in office suites selection. *European Journal of Information Systems*, 20, ss. 542-559.

Benlian, A. Hess, T. & Buxmann, P. (2009). Drivers of SaaS-adoption: an empirical study of different application types. *Business & Information Systems Engineering*, 1(5), ss. 357-369.

Bhattacharjee, A. & Park, S. C. (2014). Why end-users move to the cloud: a migration-theoretic analysis. *European Journal of Information Systems*, 23, ss. 357-372.

Bragge, J. & Merisalo-Rantanen, H. (2009). Engineering E-Collaboration Processes to Obtain Innovative End-User Feedback on Advanced Web-Based Information Systems. *Journal of the Association for Information Systems*, 10, ss. 196-220.

Damodaran, L. (1996). User involvement in the systems design process – a practical guide for users. *Behaviour & Information Technology*, 15(6), ss. 363-77.

Dubey, A. & Wagle, D. (2007). Delivering software as a service. *The McKinsey Quarterly*, Web exclusive, may 2007.

Tillgänglig: http://www.ilabsolutions.com/s/Delivering_software_as_a_service.pdf [2015-05-11]

Heller, F., Lichtschlag, L., Wittenhagen, M., Karrer, T. & Borchers, J. (2011). Me hates this: exploring different levels of user feedback for (usability) bug reporting. I *Proceedings of the 2011 annual conference extended abstracts on Human factors in computing systems (CHI EA '11)*. ACM, New York, NY, USA, ss. 1357-1362.

Ives, B. & Olson, M.H. (1984). User Involvement and MIS Success: A Review of Research. *Management Science*, 30(5), ss. 586-603.

- Kujala, S. (2003). User involvement: a review of the benefits and challenges. *Behaviour and Information Technology*, 22(1), ss. 1-17.
- Lohmann, S. & Rashid, A. (2008) Fostering Remote User Participation and Integration of User Feedback into Software Development. I *Proceedings of I-Used'08*. Pisa, Italy 24 September 2008, ss. 1-3.
- Lynch, T. & Gregor, S. (2004). User participation in decision support systems development: influencing system outcomes. *European Journal of Information Systems*, 13, ss. 286-301.
- Magnusson, J. & Nilsson, A. (2014). *Enterprise system platforms: transforming the agenda*. 1. uppl. Lund: Studentlitteratur.
- McKeen, J. D., Guimaraes T. & Wetherbe J.C. (1994). The Relationship between User Participation and User Satisfaction: An Investigation of Four Contingency Factors. *MIS Quarterly*, 18(4), ss. 427-451.
- Newman, M. & Noble, F. (1990). User involvement as an Interaction Process: A case study. *Information Systems Research*, 1, ss. 89-113.
- Nichols, D., McKay, D. & Twidale, M. (2003). Participatory usability: supporting proactive users. I *Proceedings of the 4th Annual Conference of the ACM Special Interest Group on Computer-Human Interaction (CHINZ '03)*. ACM, New York, NY, USA, ss. 63-68. DOI:10.1145/2331829.2331841
- Nichols, D., & Twidale, M. (2003). The Usability of Open Source Software. *First Monday*, 8(1). Tillgänglig: <http://firstmonday.org/ojs/index.php/fm/article/view/1018> [2015-05-11]
- Norman, D. (2002[1988]). *The Design of Everyday Things*. First Basic paperback ed. New York: Basic Books.
- Pagano, D. & Bruegge, B. (2013). User Involvement in Software Evolution Practice: A Case Study. I *Proceedings of the 35th International Conference on software engineering*. San Francisco, CA, USA, ss. 953-962.
- Papazoglou, M.P & Ribbers, P.M.A. (2006). *E-Business - Organizational and Technical Foundations*. Chichester: John Wiley & Sons, Ltd.
- Patel, R. & Davidson, B. (2011). *Forskningsmetodikens grunder: att planera, genomföra och rapportera en undersökning*. 4. uppl., Lund: Studentlitteratur.

Ramler, R., Wolfmaier, K & Weippl, E. (2004). From Maintenance to Evolutionary Development of Web Applications: A Pragmatic Approach. I *Proceedings of the International Conference on Web Engineering*, Munich, Germany, Springer-Verlag, ss. 287-299.

Schneider, K. (2011) Focusing spontaneous feedback to support system evolution. I *Requirements Engineering Conference (RE)*. Augusti 29-2 September 2011 ss. 165-174. DOI: 10.1109/RE.2011.6051645

Sein, M.K., Henfridsson, O., Purao, S., Rossi, M. & Lindgren, R. (2011). Action Design Research. *MIS Quarterly*, 35(1), ss. 37-56.

Sharp, H., Rogers, Y. & Preece, J. (2011) *Interaction Design – Beyond Human-Computer Interaction*. 3. uppl., Chichester: John Wiley & Sons, Ltd.

Tidwell, J. (2011). *Designing Interfaces – Patterns for effective Interaction Design*. 2. uppl., Sebastopol: O'Reilly Media, Inc.

Vella, J. (2011) *Determinants in the Adoption of a Customer Relationship Management System: A Study among Bank Managers*, Diss. Luleå Tekniska Universitet, Sverige.

Verma , G. (2011). Software as a Service. *Pragyaan: Journal of Information Technology*, 9(1), ss. 32-37.

Wu, W.-W. (2011). Developing an explorative model for SaaS adoption. *Expert Systems with Applications*, 38(12), ss. 15057-15064.

Bilaga 1

Intervjuguide intervju 1, informant 1

Allmänt

Vilken roll har du?

Kan du beskriva ert system?

Hur länge har systemet varit i bruk?

Hur många använder sig av systemet idag?

Användarna

Vilka typer av användare är det som använder systemet?

Hur upplever ni användarna? Grad av IT-kunnighet.

Hur ser kommunikationen ut med användarna ut? Hur sker den?

Feedback

Hur ser feedbacken ut? I vilka kanaler kommer feedbacken in?

Hur motiverar ni användarna till att ge feedback?

Vilken typ av feedback får ni in? - vilken typ VILL ni ha in?

Vad är bra, vad är dålig feedback?

Hur hanteras feedbacken nu?

Vilka typer av användare är det som ger feedback?

Har ni svårt att förstå varandra när ni pratar kring tekniken? Hur yttrar sig det?

Ger ni någon återkoppling till användarna på deras feedback?

Meddelar ni användarna när ni gjort något i systemet? I så fall hur då?

Gör ni någon skillnad på om förslag kommer från en ny kund eller en trogen kund?

Bilaga 2

Intervjuguide telefonintervju 1, informant 1 och 2

Allmänt

Vad har du för yrkesroll?

Systemet (när vi pratar om systemet pratar vi om webb-tjänster och inte mobil-appen)

Hur länge har du använt systemet Struqtur?

Vad använder du systemet till?

Feedback

Har du upplevt några problem med systemet?

I så fall vilka?

Har du kontaktat support angående det?

- Vid ja-svar: Hur skedde det?
- Vid nej-svar: varför inte?

Har du kommit på några förbättringar på systemet?

Vad gällde förbättringarna?

Har du kontaktat support angående det?

- Vid ja-svar: Hur skedde det?
- Vid nej-svar: varför inte?

Varför valde du att ringa/maila istf ringa/maila?

Hur upplever du kontakten med Struqturs support?

Hur ställer du dig till en feedback-funktion i systemet som du själv fyller i?

- Ev. följdfrågor: vid vagt svar. kan du se några nackdelar/fördelar med det?

Hur tar du del av nyheter i systemet?

Bilaga 3

Evalueringsteman för iteration 2 och 3 med informant 1

- Visuellt
 - Ikoner
 - Annat
- Information
 - Vad som visas
 - Vad som hämtas
- Användarvänlighet
 - För Struktur
 - För användarna
- Funktionalitet
 - Drag and Drop
 - Aktuella ärenden
 - Formuläret

Bilaga 4

Evalueringsinstruktion för informant 2 och 3

Vi har satt upp en testmiljö av Struqtur. Vi har tagit förstasidan i systemet inloggade som en användare i det påhittade företaget "Bygg AB". Det finns en begränsad funktionalitet i denna testmiljö och det går inte att klicka runt i systemet utan det är bara denna vy som finns tillgänglig.

Vi har utformat tre uppgifter som är tänkt att simulera verkliga händelser.

Klicka gärna runt bland de olika valen i feedbackfunktionen och "lek" med den innan testandet av scenarierna presenterade här nedan.

Fundera gärna kring dessa eller andra saker ni kommer på under testet.

- Tydligheten i instruktionerna
- Är funktionen lätt att använda
- Skulle du använda dig av funktionen om den implementerades i systemet
- Någon funktion du saknar
- Är symbolerna tydliga
- Vad är bra och vad kan vi göra bättre

Uppgift 1:

Rapportera in att byggnyheter inte fungerar.

Uppgift 2:

Det hade varit bra med en rensa-knapp vid textinmatningen, ge det som förslag.

Uppgift 3:

Systemet går segt i allmänhet. Rapportera det.

Bilaga 5

Intervjufrågor evaluering med informant 2 och 3

Hur gick du tillväga för att lösa uppgift 1?

Hur gick du tillväga för att lösa uppgift 2?

Hur gick du tillväga för att lösa uppgift 3?

Vad tyckte du om vår lösning?

Stötte du på något problem, Vad då?

Upplevde du att funktionen var lätt att hitta?

Tyckte du det var att förstå hur du stängde ner funktionen?

Hur var det att använda "drag and drop"-funktionen?

Märkte du av att du fick ett svar på den feedback du skickat in?

Om så, vad tycker du om den funktionen?

Skulle du använda dig av denna funktionen om den fanns i systemet?

I vilka lägen skulle det vara i så fall?

Hur upplever du att lämna feedback på det här viset kontra att ringa?

Vad var bra och vad kan bli bättre?

Bilaga 6

Intervjuförfrågan

Hej Informant,

Vi är två studenter som studerar tredje och sista året systemvetenskap på Göteborgs universitet och håller på med vårt examensarbete. Vi samarbetar med Struqtur för att utvärdera och förbättra feedback-processen i systemet. Vi har fått dina kontaktuppgifter av Struqtur.

Tanken är att vi ska bygga en ny feedback-funktion i systemet och vi skulle gärna ha dina åsikter kring detta så att vi får med användarnas perspektiv.

Vi vill med intervjun undersöka hur situationen ser ut och kommer ställa några frågor (som vi skickar till dig innan) som berör systemet och användningen av det. Vi skulle också vilja att du var med och utvärderade vår lösning genom att testa den som också eventuellt dokumenteras via en intervju. Vårt arbete kommer att publiceras som en vetenskaplig artikel och kommer vara offentlig. Du kommer att vara anonym och kommer endast benämnas som en "informant" tillsammans med din yrkesroll i vårt arbete. Vi skulle vilja spela in intervjun/intervjuerna för att kunna återge den/dem på ett korrekt sätt. Alla inspelningar kommer naturligtvis raderas när arbetet är klart.

Det skulle betyda mycket för oss om du ville ställa upp på detta även om det bara är delar av det eller att du vill att det ska ske på ett annat vis än det som vi föreslagit här.

Är detta något du skulle kunna tänka dig att göra?